

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

На правах рукопису
УДК 004.056.53

До захисту допущено
В. о. завідувача кафедри ММСА
О.Л.Тимошук
«___» _____ 2020 р.

Магістерська дисертація

на здобуття ступеня магістра за спеціальністю 124 Системний аналіз
на тему: «Генеративно-змагальна нейронна мережа для прогнозування
фондового ринку»

Виконала:

студентка II курсу, групи КА-92 мп
Бородай Тетяна Анатоліївна

Керівник:

професор кафедри ММСА
д.т.н. проф. Данилов В.Я.

Рецензент:

професор кафедри інформаційної безпеки
д.т.н. проф. Качинський А.Б.

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів
без відповідних посилань

Студентка _____

Київ
2020

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Рівень вищої освіти — другий (магістерський)
Спеціальність — 124 «Системний аналіз»

ЗАТВЕРДЖУЮ

В. о. завідувача кафедри ММСА
О. Л. Тимошук

«___» _____ 2020 р.

ЗАВДАННЯ

на магістерську дисертацію студентки Бородай Тетяни Анатоліївни

1. Тема дисертації: «Генеративно-змагальна нейрона мережа для прогнозування фондового ринку», науковий керівник дисертації Данилов Валерій Якович, д.т.н. проф., затверджені наказом по університету від «02» листопада № 3182-с

2. Термін подання студентом дисертації: 14 грудня 2020 р.

3. Об'єкт дослідження: рух цін акцій.

4. Предмет дослідження: математичні методи та моделі опису прогнозування, оцінювання та аналізу якості прогнозів.

5. Перелік завдань, які потрібно розробити:

1) розглянути методи прогнозування фондових ринків;
2) розробити математичну модель для прогнозування фінансового ринку;
3) виконати обчислювальні експерименти стосовно моделювання та прогнозування фінансового ринку з використанням GANs;

4) дати оцінку і практичне застосування отриманих результатів, сформулювати наукову новизну та перспективи виконання подальших досліджень.

5) розробити стартап-проект виведення на ринок результатів дослідження;

6. Орієнтовний перелік графічного (ілюстративного) матеріалу:

1). Схема GAN (рис.);
2). Приклади функціонування створеного програмного продукту (рис.);
3). Таблиці у розділі стартап-проекту

7. Дата видачі завдання: 05 вересня 2020 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації
1.	Концептуальний вступ дисертації. Формулювання об'єкта, предмета, цілі, завдань, новизни, практичної значущості результатів	05.09.2020—12.09.2020
2.	Перший розділ. Огляд літературно-інформаційних джерел. Понятійно-категоріальний апарат. Характеристика об'єкта	16.09.2020—27.09.2020
3.	Другий розділ. GAN та її використання.	30.09.2020—19.10.2020
4.	Третій розділ. Пошук даних та реалізація алгоритму. Імплементация отриманих результатів у програмний продукт. Тестування програми	22.10.2020—15.11.2020
5.	Четвертий розділ. Стартап-проект	18.11.2020—20.11.2020
6.	Концептуальні висновки. Перспективи розвитку отриманих рішень	21.11.2020—25.11.2020

Студентка

Бородай Т.А.

Науковий керівник дисертації

Данилов В.Я.

РЕФЕРАТ

Магістерська дисертація: 106 с., 22 табл., 19 рис. 2 дод., 37 джерел.

НЕЙРОН, НЕЙРОННА МЕРЕЖА, ГЕНЕРАТИВНО-ЗМАГАЛЬНА НЕЙРОННА МЕРЕЖА, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, ПРОГНОЗ.

Об'єкт дослідження – рух цін акцій Goldman Saches.

Мета роботи – реалізувати генеративно-змагальну нейронну мережу для прогнозування фондового ринку. Дослідити вплив значень параметрів моделі на точність моделі. Порівняти результати прогнозу методу з іншими методами прогнозу.

Методи дослідження – реалізувати генеративно-змагальну нейронну мережу і дослідити її роботу.

У роботі було запропоновано просту у використанні модель прогнозування фондового ринку. Було використано навчання з підкріпленням при оптимізації гіперпараметрів, адже фондові ринки весь час змінюються. Навіть якщо вдається навчити GAN та LSTM створювати надзвичайно точні результати, результати можуть бути дійсними лише протягом певного періоду. Це означає, що потрібно постійно оптимізувати весь процес.

Заснована на мережі глибокого навчання, ця модель досягає здатності прогнозування, що перевершує інші базові методи, за допомогою змагальних тренувань, мінімізації втрат прогнозування напрямку та втрати прогнозованих помилок.

У майбутньому ми спробуємо створити середовище RL для тестування торгових алгоритмів, які вирішують, коли і як торгувати. Вихід з GAN буде одним із параметрів у середовищі.

ABSTRACT

Master thesis: 106 p., 22 tables, 19 figures. 2 items, 37 sources.

GMDH, NEURON, NEURAL NETWORK, GENERATIVE ADVERSIAL NETWORK, FORECAST.

The object of the study is the movement of Goldman Saches shares.

The purpose of the work is to implement a hybrid generative-competitive neural network for stock market forecasting. Investigate the influence of model parameter values on model accuracy. Compare the results of the forecast method with other forecast methods.

Research methods - to implement a generative-competitive neural network and explore its work.

In paper, an easy-to-use stock market forecasting model was proposed. Reinforced training was used to optimize hyperparameters, as stock markets are constantly changing. Even if you can teach GAN and LSTM to create extremely accurate results, the results can only be valid for a certain period of time. This means that you need to constantly optimize the whole process.

Based on a network of deep learning, this model achieves the ability to predict, which surpasses other basic methods, through competitive training, minimizing the loss of directional prediction and loss of predicted errors. Moreover, the effects of model update cycles on predictability are analyzed, and experimental results show that a smaller model update cycle can obtain better prediction results.

In the future, we will try to create an RL environment for testing trading algorithms that decide when and how to trade. Exiting the GAN will be one of the parameters in the environment.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	9
ВСТУП.....	10
РОЗДІЛ 1 МЕТОДИ ПРОГНОЗУВАННЯ ФОНДОВИХ РИНКІВ ТА АКТУАЛЬНІСТЬ ДОСЛІДЖЕННЯ.....	12
1.1 Традиційні підходи.....	12
1.2 Підходи на основі застосування математичних методів	13
1.2.1 Інструменти на основі теорії детермінованого хаосу.....	13
1.2.2 Інструменти на основі розробок в галузі ШІ.....	14
1.3 Використання розробок в галузі ШІ на фондовому ринку.....	15
1.4 Огляд літератури.....	16
1.5 Ефективність застосування нейронних мереж на фондовому ринку.....	18
1.6 Постановка задачі дослідження	19
1.7 Висновки	20
РОЗДІЛ 2 ГЕНЕРАТИВНО-ЗМАГАЛЬНІ НЕЙРОННІ МЕРЕЖІ ДЛЯ ПРОГНОЗУВАННЯ	21
2.1 Генеративно-змагальна нейронна мережа.....	21
2.2 Архітектура GANs	24
2.3 Metropolis-Hastings GAN and Wasserstein GAN	26
2.4 Wasserstein GAN	27
2.5 Генератор – одношаровий RNN.....	28
2.6 Дискримінатор – одновимірний CNN	30
2.7 Навчання GAN	35
РОЗДІЛ 3 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА	38

3.1. Відповідні активи	38
3.2. Технічні показники.....	39
3.3. Фундаментальний аналіз.....	41
3.4. Перетворення Фур'є для аналізу тенденцій.....	42
3.5. ARIMA як особливість	44
3.6. Статистичні перевірки.....	47
3.7. Функціональна інженерія.....	48
3.8. Видобування високорівневих функцій за допомогою автоматичних кодерів з накопиченням	51
3.8.1. Функція активації – GELU (Гауссова помилка)	51
3.8.2. Eigen портфолію з PCA	54
3.8.3. Планувальник швидкості навчання	55
3.8.4. Як запобігти переобладнанню та компромісу між зміщенням ..	56
3.9 Гіперпараметри	58
3.10. Підсилення навчання для оптимізації гіперпараметрів.....	59
3.11 Результати роботи.....	60
3.12 Висновки до розділу 3	62
РОЗДІЛ 4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ	63
4.1 Опис ідеї проекту.....	63
4.2 Технологічний аудит проекту	66
4.3 Аналіз ринкових можливостей	67
4.4 Розробка ринкової стратегії проекту	78
4.5 Розробка маркетингової програми стартап-проекту	84
4.6 Висновки	89
ВИСНОВКИ	90

ПЕРЕЛІК ПОСИЛАНЬ	91
ДОДАТОК А ВХІДНІ ДАНІ.....	95
ДОДАТОК Б ЛІСТІНГ ПРОГРАМИ.....	101

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

1. GAN – генеративно-змагальна нейронна мережа
2. НМ – нейронна мережа
3. CNN – згорткова нейронна мережа
4. GS – Goldman Saches
5. GELU – помилка Гауса

ВСТУП

Необхідність у передбаченні змін на фондових ринках виникла з моменту появи цих ринків. Розвиток обчислювальної техніки уможливив практичну реалізацію задач прогнозування на базі напрацьованого математичного апарату. Кожне підвищення точності прогнозування забезпечує учасникам ринку не лише зменшення ризиків їхньої діяльності, а й отримання значних фінансових вигадів. Сучасні обчислювальні потужності дають змогу здійснювати передбачення на основі гігантської вибірки розрізнених та слабо зв'язаних даних при котирування цінних паперів на біржах. Серед ефективних технологій та підходів до прогнозування особлива увага за останні роки приділялася штучним нейронним мережам, які використовуються у багатьох сферах автоматизації багатокритеріальних задач. Завдяки своїм внутрішнім якостям, зокрема здатності до «запам'ятовування» інформації, та нелінійним властивостям обчислювальної одиниці – нейрону – такі мережі здатні розв'язувати задачі класифікації, здійснювати розпізнавання образів та виконувати прогнозування.

Точне прогнозування фондових ринків – складна задача, оскільки існують мільйони подій і попередніх умов для руху певних акцій в певному напрямку.

Генеративно-змагальні мережі (GAN) останнім часом використовуються головним чином для створення реалістичних зображень, картин і відеокліпів. Існує не так багато додатків GAN, використовуваних для прогнозування даних часових рядів, як в нашому випадку.

У зв'язку з цим є вкрай актуальними наукові дослідження в галузі математичного моделювання прогнозування фондових ринків з використанням генеративно-змагальних нейронних мереж (GAN).

У першому розділі розглянуто відомі методи прогнозування фондових

ринків, обґрунтовано вибір тематики дослідження, а також зроблено постановку задачі дослідження.

Другий розділ присвячено гетеративно-змагальним нейронним мережам та їх роботі у прогнозуванні фондових ринків.

У третьому розділі наведено побудову моделей та проведено аналіз отриманих результатів.

РОЗДІЛ 1 МЕТОДИ ПРОГНОЗУВАННЯ ФОНДОВИХ РИНКІВ ТА АКТУАЛЬНІСТЬ ДОСЛІДЖЕННЯ

1.1 Традиційні підходи

Вивчення проблеми прогнозування динаміки курсових вартостей акцій фондового ринку почалося з аналізу використовуваних методів та інструментів прогнозування.

Спочатку були проаналізовані традиційні підходи до прогнозування динаміки курсових вартостей акцій фондового ринку – фундаментальний і технічний аналіз.

Основний висновок проведеного аналізу, заснованого на вивченні багаторічної практики застосування даних інструментів, описаної в спеціалізованій літературі – дані підходи мають низькі прогностичні можливості, явно недостатні для забезпечення прийняттого рівня прибутковості, стабільної в довгостроковій перспективі. Не останню роль в цьому зіграла слабкість теоретичної основи традиційних підходів, небезпідставно піддавалася критиці протягом всього періоду їх існування.

Така ситуація призвела до того, що фахівці в області фондового ринку стали шукати нові можливості підвищення ефективності прогнозування. Основним напрямком розробки нових прогностичних інструментів стало застосування різних математичних методів.

1.2 Підходи на основі застосування математичних методів

Так утворилося два основних напрямки застосування математичних методів прогнозування: напрямок, заснований на теорії детермінованого хаосу і напрямок, заснований на розробках в області штучного інтелекту.

1.2.1 Інструменти на основі теорії детермінованого хаосу

Даний напрямок є досить новим і говорити про його ефективність поки, можливо, дещо передчасно. На даний же момент, аналіз спеціалізованої літератури показав, що практично всі публікації, в яких наводиться опис прогностичних інструментів на основі теорії детермінованого хаосу, відносяться до періодичної літератури. У жодному джерелі не було виявлено докладну розповідь застосування інструментів даного класу в умовах реального фондового ринку, із зазначенням результатів, які можна було б вважати достовірними і стійкими в часі. Незважаючи на те, що в публікаціях, присвячених даному напрямку, іноді можна зустріти описи позитивних результатів застосування даного підходу в прогнозуванні динаміки курсових вартостей акцій, вони мають явно упереджений характер.

Відгуки ж практиків говорять прямо про зворотне. Думки безпосередніх інвесторів фондового ринку зводяться швидше до того, що ефективність подібних інструментів досить сумнівна, і навряд чи можна стверджувати, що вона перевищує ефективність прогностичних інструментів, створених на основі традиційних підходів. Тому і розповсюдження інструментів на основі теорії

детермінованого хаосу серед інвесторів дуже мала.

1.2.2 Інструменти на основі розробок в галузі ШІ

Більш тривалу історію має підхід, заснований на застосуванні для прогнозування динаміки курсових вартостей акцій фондового ринку розробок в області ШІ.

Штучний інтелект – розділ інформатики, що вивчає можливість відтворення міркувань і дій, аналогічних мислення і дій людини, за допомогою електронних обчислювальних систем та інших штучних пристроїв. Головним завданням розробки систем штучного інтелекту є побудова штучної системи, наприклад – комп'ютерної програми, яка має рівень ефективності рішення неформалізованих завдань, який можна порівняти з людським або перевершує його.

Історично виділилося два основних підходи до побудови систем ШІ: логічний і нейрокібернетичний.

1. Логічний підхід – математичне моделювання свідомої діяльності людини, тобто моделювання високорівневих процесів мислення, заснованих на поданні та використанні знань. Методологія даного підходу – виявлення і застосування в інтелектуальних системах різних логічних і емпіричних прийомів (евристик), які людина використовує для вирішення будь-яких завдань. З розвитком на початку 80-х років концепції експертних систем цей напрямок вилився в науково-технологічний напрямок інформатики «інженерія знань», що займається створенням так званих «систем, заснованих на знаннях». Алгоритми вирішення задач, використовуваних в таких системах, закладаються експертами,

у вигляді аксіоматичних правил.

2. Нейрокібернетичний (нейромережевий) підхід – заснований на математичному моделюванні функціонування низькорівневих структур мозку – нейронів. Його методологія – побудова систем, що самоорганізуються, що складаються з безлічі елементів, функціонально подібних нейронів головного мозку. Нейрокібернетичний напрямок відносять до моделювання мислення на підсвідомому рівні (Моделювання інтуїції, творчої уяви, інсайту). Основний внесок в розвиток цього підходу внесли вчені країн Заходу і США.

В цілому, теоретична база підходу до розробки систем штучного інтелекту на сьогоднішній день вкрай слабка. Не визначено багато базові поняття, до сих пір не досягнуто згоди навіть у визначенні того, що саме вважати інтелектом. Тому дослідження в цій області є відкритими.

Сфери застосування розробок в області ШІ вкрай різноманітні: медицина, геологія, військова техніка, економіка та ін. Такі розробки використовуються для вирішення задач передбачення, діагностики, планування, управління. Нейронні мережі здатні вирішувати завдання в умовах неповноти, спотворення, зашумленості інформації, а також при відсутності алгоритму вирішення задач.

1.3 Використання розробок в галузі ШІ на фондовому ринку

Завдання прогнозування динаміки курсових вартостей акцій фондового ринку відноситься до неформалізованих, тому для її вирішення використовуються розробки в області нейромережевого підходу – нейронні мережі.

На сьогоднішній день такі нейромережеві пакети, отримали досить широке

поширення на фондовому ринку, особливо в країнах Заходу. Серед найбільш відомих програмних продуктів можна назвати: NeuroShell Day Trader, Neuro Builder, Brain Maker, NeuroScalp.

На західному фондовому ринку нейронні мережі стали використовуватися ще в 90-х роках, з початком різкого розвитку комп'ютерних технологій. На українському фондовому ринку вони з'явилися всього кілька років тому, що пов'язано в першу чергу з розвитком інфраструктури фондового ринку.

1.4 Огляд літератури

Було розглянуто безліч статей про GANs та області їх використання. Можна зробити висновок, що на сьогоднішній день GANs здебільшого використовуються в системах комп'ютерного зору. Проте, нещодавно спробували використовувати GANs в задачах прогнозування.

Згідно з дослідженнями, розробленим в цій області, ми можемо розділити методи, використовувані для вирішення завдань прогнозування фондового ринку, на дві групи.

Перша категорія пов'язаних робіт – це економетричні моделі, які включають класичні економетричні моделі для прогнозування. Поширеними методами є метод авторегресії (AR), модель змінного середнього (MA), модель авторегресійного змінного середнього (ARMA) і авторегресійна інтегрована змінна середня (ARIMA) [17 - 19]. Грубо кажучи, ці моделі сприймають кожен новий сигнал як зашумленну лінійну комбінацію декількох останніх сигналів і незалежних шумових складових. Однак більшість з них покладаються на деякі сильні припущення щодо шумових складових і функції втрат, тоді як реальні

фінансові дані можуть не повністю відповідати цим припущенням. Шляхом введення узагальненої авторегресійної умовно-гетероскедастичної моделі (GARCH) для умовних дисперсій Pellegrini et al. [20] застосовують модель ARIMA-GARCH для прогнозування фінансових часових рядів.

Друга категорія включає моделі, засновані на м'яких обчисленнях. М'які обчислення – це термін, що позначає штучний інтелект, що імітує біологічні процеси. Ці методи включають штучні нейронні мережі (ШНМ) [21, 22], нечітку логіку (FL) [23], допоміжні векторні машини (SVM) [24, 25], оптимізацію рою частинок (PSO) [26] і багато інших. Багато авторів намагалися мати справу з нечіткістю і випадковістю в моделях ціноутворення опціонів [27, 28]. Карлссон і Фуллер [29] були першими, хто вивчив нечіткі реальні варіанти, а Таванесваран і ін. [30] продемонстрував перевагу нечітких прогнозів, а потім вивів функцію приналежності для європейської ціни шляхом нечіткого визначення відсоткової ставки, волатильності і початкового значення ціни акцій.

Останнім часом спостерігається відродження інтересу до глибокого навчання, базову структуру якого найкраще описати як багатоваршівну нейронну мережу [31]. У деяких літературних джерелах створені різні моделі, засновані на глибоких нейронних мережах, для поліпшення передбачуваної здатності високочастотних фінансових часових рядів [32]. Здатність глибоких нейронних мереж витягувати абстрактні функції з даних також приваблива, Chong et [12] застосувала модель прогнозування фондового ринку, засновану на глибокому навчанні, яка витягує інформацію з часових рядів прибутковості акцій, не покладаючись на попередні знання предикторів, і перевіряє її на високочастотних даних корейського фондового ринку. Chen [34] запропонували двошарову нейронну мережу для високочастотного прогнозування зі зв'язками, спеціально розробленими для виявлення структур залежності між прибутковістю акцій в різних секторах бізнесу. Також існує кілька досліджень, в яких глибоке навчання

застосовується для визначення взаємозв'язку між минулими новинними подіями і рухами фондового ринку [35 - 37].

Однак, наскільки нам відомо, для більшості цих методів потрібен досвід, щоб накласти певні обмеження на вхідні змінні, такі як об'єднання пов'язаних акцій разом в якості вхідних даних [12], введення різних індексних даних в різні рівні глибокої нейронної мережі [34], і перетворення тексту новин в структуроване уявлення в якості вхідних даних [36].

1.5 Ефективність застосування нейронних мереж на фондовому ринку

На практиці нейронні мережі, реалізовані в стандартних спеціалізованих програмних пакетах, мають досить низьку прогностичну ефективність. Незважаючи на більше число рекламних заяв про успішність їх застосування на фондовому ринку, реальна практика, як правило, показує зворотні результати. Так, наприклад, найбільш успішні інвестори, як правило, не рекомендують їх використання, посиляючись саме на їх неефективність.

1.6 Постановка задачі дослідження

Точне прогнозування фондових ринків – складна задача, оскільки існують мільйони подій і попередніх умов для руху певних акцій в певному напрямку. Таким чином, ми повинні охопити якомога більше таких попередніх умов. Нам також необхідно зробити декілька важливих припущень:

1. Ринки не являються випадковими на 100%.
2. Історія повторюється.
3. Ринки слідує за раціональною поведінкою людей.
4. Ринки являються «ідеальними».

Для досягнення поставленої мети дослідження в роботі послідовно вирішуються наступні задачі:

- Розглянути методи прогнозування фондових ринків.
- Розробити математичну модель для прогнозування фінансового ринку.
- Виконати обчислювальні експерименти стосовно моделювання та прогнозування фінансового ринку з використанням GANs.
- Дати оцінку і практичне застосування отриманих результатів, сформулювати наукову новизну та перспективи виконання подальших досліджень.

1.7 Висновки

Проведений аналіз показав, що використовувані на фондовому ринку прогностичні інструменти не мають прийнятного рівня прогностичної сили.

Така ситуація обумовлена і досить слабкою теоретичною основою прогнозування фінансових інструментів фондового ринку. Це залишає проблему розробки нового теоретико-методологічного підходу до прогнозування динаміки курсових вартостей акцій фондового ринку актуальною.

Найбільш перспективним напрямком створення більш ефективних прогностичних інструментів представляється ведення розробок в області ІІІ.

Наявні розробки в цій області – нейромережі вже володіють деякими потенційними перевагами перед традиційними підходами та іншими математичними підходами. Наприклад, вони дозволяють враховувати найбільшу кількість факторів, здатні вирішувати завдання, що не мають алгоритму та ін. Однак, нейромережі володіють і істотними недоліками, які практично зводять нанівець їх гідності, наприклад, відсутність безперервного перенавчання нейромережі в процесі виконання завдання досить швидко робить її неадекватною ситуації на ринку.

РОЗДІЛ 2 ГЕНЕРАТИВНО-ЗМАГАЛЬНІ НЕЙРОННІ МЕРЕЖІ ДЛЯ ПРОГНОЗУВАННЯ

2.1 Генеративно-змагальна нейронна мережа

Що означає «генеративна» в назві «Генеративно-змагальна мережа»? «Генеративний» описує клас статистичних моделей, який контрастує з дискримінаційними моделями.

Генеративні моделі можуть генерувати нові екземпляри даних.

Дискримінаційні моделі розрізняють різні типи примірників даних.

Генеративна модель може створювати нові фотографії тварин, які виглядають як справжні тварини, в той час як відмінна модель може відрізнити собаку від кішки. GAN – це всього лише один з видів генеративної моделі.

Більш формально, з огляду на набір екземплярів даних X і набір міток Y :

Генеративні моделі фіксують спільну ймовірність $p(X, Y)$ або просто $p(X)$, якщо немає міток.

Дискримінативність моделі фіксують умовну ймовірність $p(Y | X)$.

Генеративна модель включає в себе розподіл самих даних і показує, наскільки вірогідний даний приклад. Наприклад, моделі, які пророкують наступне слово в послідовності, зазвичай є генеративними моделями (зазвичай набагато простіше, ніж GAN), оскільки вони можуть призначати ймовірність послідовності слів.

Дискримінантна модель ігнорує питання про те, чи можливий цей екземпляр, і просто говорить вам, наскільки ймовірно, що мітка буде застосовуватися до примірника.

Це дуже загальне визначення. Є багато видів генеративних моделей. GAN – це лише один з видів генеративної моделі.

Жодна з моделей не повинна повертати число, яке представляє ймовірність. Можна змодельовати розподіл даних, імітуючи цей розподіл.

Наприклад, дискримінантний класифікатор, такий як дерево рішень, може помітити екземпляр без присвоєння цій мітці ймовірності. Такий класифікатор все одно буде моделлю, тому що розподіл всіх передбачених міток буде моделювати реальний розподіл міток в даних.

Точно так генеративна модель може моделювати розподіл, виробляючи переконливі «фальшиві» дані, які виглядають так, як ніби вони взяті з цього розподілу.

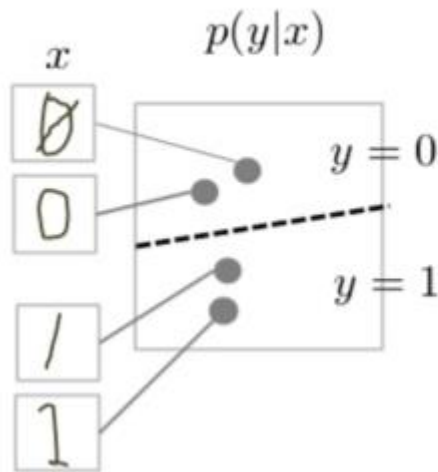
Генеративні моделі вирішують більш складне завдання, ніж аналогічні дискримінантні моделі. Генеративні моделі повинні моделювати більше.

Генеративна модель зображень може вловлювати кореляції на кшталт «речі, схожі на човни, ймовірно, будуть з'являтися поряд з речами, які виглядають як вода» і «очі навряд чи з'являться на лобі». Це дуже складні дистрибутиви.

На відміну від цього, дискримінаційна модель може дізнатися різницю між «парусної човном» і «не парусної човном», просто шукаючи кілька характерних закономірностей. Він може ігнорувати багато кореляції, які генеративна модель повинна встановлювати правильно.

Дискримінаційні моделі намагаються окреслити межі в просторі даних, в той час як генеративні моделі намагаються змодельовати, як дані розміщуються в просторі. Наприклад, на рисунку 2.1 показані відмітні і породжують моделі рукописних цифр:

- Discriminative Model



- Generative Model

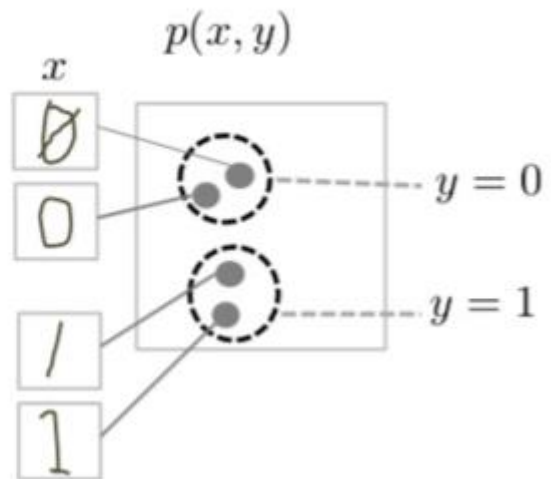


Рисунок 2.1 – схеми генеративної та дискримінативної моделей

Дискримінантна модель намагається визначити різницю між написаними від руки 0 і 1, малюючи лінію в просторі даних. Якщо він правильно побудує рядок, він зможе відрізнити 0 від 1 без необхідності точно моделювати, де екземпляри розміщуються в просторі даних по обидва боки від рядка.

Навпаки, генеративна модель намагається створити переконливі одиниці і нулі, генеруючи цифри, які близькі до їх реальним аналогам в просторі даних. Він повинен моделювати розподіл в просторі даних.

Мережі GAN пропонують ефективний спосіб навчання таких багатих моделей, щоб вони нагадували реальний розподіл. Щоб зрозуміти, як вони працюють, нам потрібно зрозуміти базову структуру GAN.

2.2 Архітектура GANs

Мережа GAN складається з двох моделей – генератора (G) та дискримінатора (D). На рисунку 2.2 зображено архітектуру GAN.

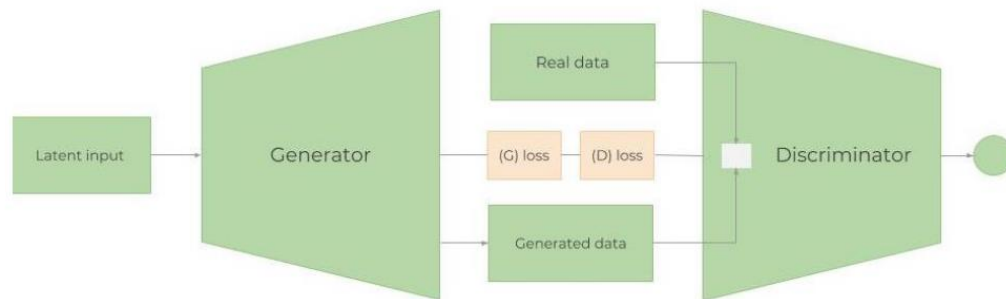


Рисунок 2.2 – Архітектура GAN

Етапи навчання GAN:

Генератор, використовуючи випадкові дані (шум, позначений z), намагається «генерувати» дані, які не відрізняються від реальних даних або надзвичайно близькі до них. Його мета – навчитися розподілу реальних даних. Випадково реальні або сформовані дані вкладаються в дискримінатор, який діє як класифікатор і намагається зрозуміти, чи надходять дані від генератора, чи це реальні дані. D оцінює (розподіли) ймовірності вхідної вибірки до реального набору даних.

Потім втрати від G і D об'єднуються і передаються назад через генератор. Втрати генератора залежать як від генератора, так і від дискримінатора. Це крок, який допомагає Генератору дізнатися про реальний розподіл даних. Якщо генератор не справляється добре з генерацією реалістичних даних (з однаковим розподілом), Дискримінатору буде дуже легко відрізнити згенеровані дані від реальних наборів даних. Отже, втрата дискримінатора буде дуже невеликою. Невеликі втрати дискримінатора

призведуть до більших втрат генератора (див. рівняння нижче для $L(D, G)$). Це робить створення дискримінатора трохи складним, оскільки занадто хороший дискримінатор завжди призведе до величезних втрат генератора, через що генератор не зможе вчитися.

Процес триває доти, доки Дискримінатор більше не зможе відрізнити генеровані дані від реальних даних.

Поєднання D і G є різновидом гри в minmax (Генератор намагається обдурити Дискримінатор, роблячи так, щоб він збільшував ймовірність на підроблених прикладах, тобто мінімізував $\mathbb{E}_{z \sim p_z(z)} [-\log(1 - D(G(z)))]$. Дискримінатор хоче відокремити дані, що надходять від Генератора, $D(G(z))$, максимізуючи $\mathbb{E}_{x \sim p_r(x)} [-\log D(x)]$. Однак, розділивши функції збитків, незрозуміло, як обидві можуть сходитися разом (саме тому ми використовуємо певний прогрес порівняно із звичайними GAN, такими як Wasserstein GAN). Загалом, функція комбінованих втрат виглядає так:

$$L(D, G) = \mathbb{E}_{x \sim p_r(x)} [-\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [-\log(1 - D(G(z)))]$$

Чому ми використовуємо GAN для прогнозування на фондовому ринку?

Генеративно-змагальні нейронні мережі (GAN) нещодавно використовуються головним чином для створення реалістичних зображень, картин та відеокліпів. Не так багато застосувань GAN, які використовуються для прогнозування даних часових рядів, як у нашому випадку. Однак головна ідея повинна бути однаковою - ми хочемо передбачити майбутні руху акцій. У майбутньому структура і поведінка акцій GS повинні бути більш-менш однаковими (якщо тільки вона не почне працювати зовсім по-іншому, або економіка не зміниться кардинально). Отже, ми хочемо „генерувати” дані на майбутнє, які матимуть подібний (не абсолютно однаковий, звичайно) розподіл, як той, який ми вже маємо - історичні торгові дані. Отже, теоретично це має

спрацювати.

У нашому випадку ми будемо використовувати LSTM як генератор часових рядів, а CNN як дискримінатор.

2.3 Metropolis-Hastings GAN and Wasserstein GAN

Нещодавнє вдосконалення традиційних GAN вийшло від інженерної команди Uber і називається Metropolis-Hastings GAN (MHGAN). Ідея підходу Uber (як вони заявляють) дещо схожа на інший підхід, створений Google та Каліфорнійським університетом, Берклі, під назвою Discriminator Rejection Sampling (DRS). В основному, коли ми навчаємо GAN, ми використовуємо дискримінатор (D) з єдиною метою кращої підготовки генератора (G). Часто після навчання GAN ми більше не використовуємо D . Однак MHGAN і DRS намагаються використовувати D , щоб вибрати зразки, сформовані G , які близькі до реального розподілу даних (невелика різниця між тим, що MHGAN використовує ланцюг Маркова Монте-Карло (MCMC) для вибірки).

MHGAN бере K зразки, згенеровані з G (створені з незалежних входів шуму в G - z_0 до z_K зображено на рисунку 2.3 нижче). Потім він послідовно проходить через виходи $K(x'_0$ до $x'_K)$ і, слідуючи правилу прийняття (створеному з дискримінатора), приймає рішення, прийняти поточний зразок або зберегти останній прийнятий. Останній збережений випуск - той, який вважається реальним обсягом G .

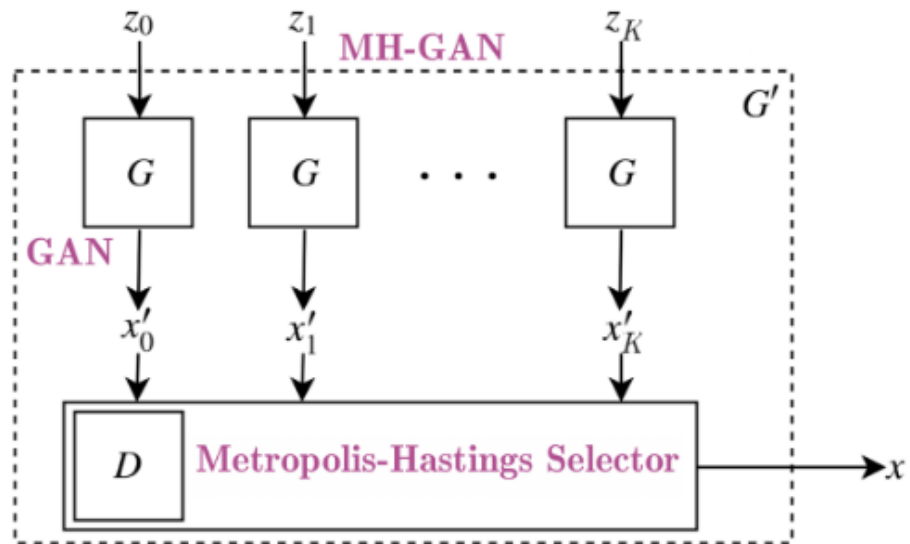


Рисунок 2.3 – Візуальне представлення MHGAN (з оригінального повідомлення Uber)

2.4 Wasserstein GAN

Навчання GANs досить складно. Моделі можуть ніколи не сходитися. Ми будемо використовувати модифікацію GAN під назвою Wasserstein GAN - WGAN.

Головна мета, що стоїть за GAN, полягає в тому, щоб генератор почав перетворювати випадковий шум у деякі дані, які ми хочемо імітувати. Ідея порівняння подібності між двома розподілами дуже необхідна в GAN. Двома найбільш широко використовуваними такими показниками є:

Розбіжність KL (Kullback – Leibler) - $D_{\{KL\}}(p|q) = \int x p(x) \log \frac{p(x)}{q(x)} dx$. $D_{\{KL\}}$ дорівнює нулю, коли $p(x)$ дорівнює $q(x)$,

JS Дивергенція (Йенсен – Шеннон) - $D_{\{JS\}}(p|q) = \frac{1}{2} D_{\{KL\}}(p|q) + \frac{1}{2} D_{\{KL\}}(q|p)$

$\frac{1}{2} D_{KL}(p \parallel \frac{p+q}{2}) + \frac{1}{2} D_{KL}(q \parallel \frac{p+q}{2})$. Розбіжність JS обмежена 0 і 1, і, на відміну від розбіжності KL, симетрична і плавна. Значного успіху в навчанні GAN було досягнуто, коли втрата була переведена з KL на дивергенцію JS.

WGAN використовує відстань Вассерштайна, $W(p_r, p_g) = \frac{1}{K} \sup_{\{f \mid L \leq K\}} \mathbb{E}_{x \sim p_r} [f(x)] - \mathbb{E}_{x \sim p_g} [f(x)]$ (де \sup стоїть для супремуму), як функція втрат (оскільки вона зазвичай трактується як переміщення однієї купи, скажімо, піску до іншої, причому обидві купи мають різний розподіл ймовірностей, використовуючи мінімум енергії під час перетворення). Порівняно з розбіжностями KL та JS, метрика Вассерштайна дає плавний показник (без різких стрибків розбіжностей). Це робить його набагато більш придатним для створення стабільного навчального процесу під час градієнтного спуску.

Крім того, порівняно з KL та JS, відстань Вассерштайна майже скрізь диференціюється. Як ми знаємо, під час зворотного розмноження ми диференціюємо функцію втрат, щоб створити градієнти, які, в свою чергу, оновлюють ваги. Тому наявність диференційованої функції втрат є досить важливою.

2.5 Генератор – одношаровий RNN

Як вже згадувалося раніше, генератор – це мережа LSTM, тип рекурентної нейронної мережі (RNN). RNN використовуються для даних часових рядів, оскільки вони відстежують всі попередні точки даних і можуть фіксувати закономірності, що розвиваються з часом. Завдяки своїй природі, RNN багато

часу страждають від зникаючого градієнта – тобто зміни, які отримують ваги під час тренування, стають настільки малими, що вони не змінюються, роблячи мережу нездатною сходитися до мінімальних втрат (протилежна проблема також може спостерігати часом – коли градієнти стають занадто великими. Це називається градієнтним вибухом, але рішення цього досить просте – градієнти кліпів, якщо вони починають перевищувати деяке постійне число, тобто відсікання градієнта). Дві модифікації вирішують цю проблему – блокований рекуррентний блок (GRU) та довгострокова пам'ять (LSTM). Найбільші відмінності між ними:

1. GRU має 2 ворота (оновлення та скидання), а LSTM має 4 (оновлення, введення, забуття та вихід).
2. LSTM підтримує стан внутрішньої пам'яті, тоді як GRU цього не робить.
3. LSTM застосовує нелінійність (сигмоподібну) перед вихідними воротами, GRU – ні.

У більшості випадків LSTM і GRU дають подібні результати з точки зору точності, але GRU набагато менше обчислювальних, оскільки GRU має набагато менше навчальних параметрів. Однак LSTM і набагато більше використовуються.

Строго кажучи:

$$g_t = \tanh(X_t W_{xg} + h_{t-1} W_{hg} + b_g),$$

$$i_t = \sigma(X_t W_{xi} + h_{t-1} W_{hi} + b_i),$$

$$f_t = \sigma(X_t W_{xf} + h_{t-1} W_{hf} + b_f),$$

$$o_t = \sigma(X_t W_{xo} + h_{t-1} W_{ho} + b_o),$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t,$$

$$h_t = o_t \odot \tanh(c_t),$$

де \odot - елементний оператор множення, а для всіх $x = [x_1, x_2, \dots, x_k]^T \in \mathbb{R}^k$ дві функції активації ;,

$$\sigma(x) = \left[\frac{1}{1 + \exp(-x_1)}, \dots, \frac{1}{1 + \exp(-x_k)} \right]^T$$

$$\text{tanh}(x) = \left[\frac{1 - \exp(-2x_1)}{1 + \exp(-2x_1)}, \dots, \frac{1 - \exp(-2x_k)}{1 + \exp(-2x_k)} \right]^T$$

2.6 Дискримінатор – одновимірний CNN

Чому CNN як дискримінатор?

Зазвичай ми використовуємо CNN для роботи, пов'язаної із зображеннями (класифікація, вилучення контексту тощо). Вони дуже потужно витягують об'єкти з об'єктів з об'єктів і т. Д. Наприклад, на зображенні собаки перший згортковий шар виявлятиме краї, другий почне виявляти кола, а третій – ніс. У нашому випадку точки даних формують невеликі тенденції, малі тенденції формують більші, тенденції в свою чергу формують шаблони. Здатність CNN виявляти особливості може бути використана для отримання інформації про закономірності в русі цін на акції GS.

Ще однією причиною використання CNN є те, що CNN добре працюють з просторовими даними – тобто точки даних, які знаходяться ближче один до одного, більше пов'язані між собою, ніж точки розподілу. Це має бути справедливим для даних часових рядів. У нашому випадку кожна точка даних (для кожної функції) призначена для кожного дня. Природно припустити, що чим ближче два дні один до одного, тим більше вони пов'язані між собою. Однак слід врахувати (хоча це не висвітлено в цій роботі) – це сезонність і те, як це може змінити роботу CNN. Архітектуру CNN зображено на рисунку 2.4.

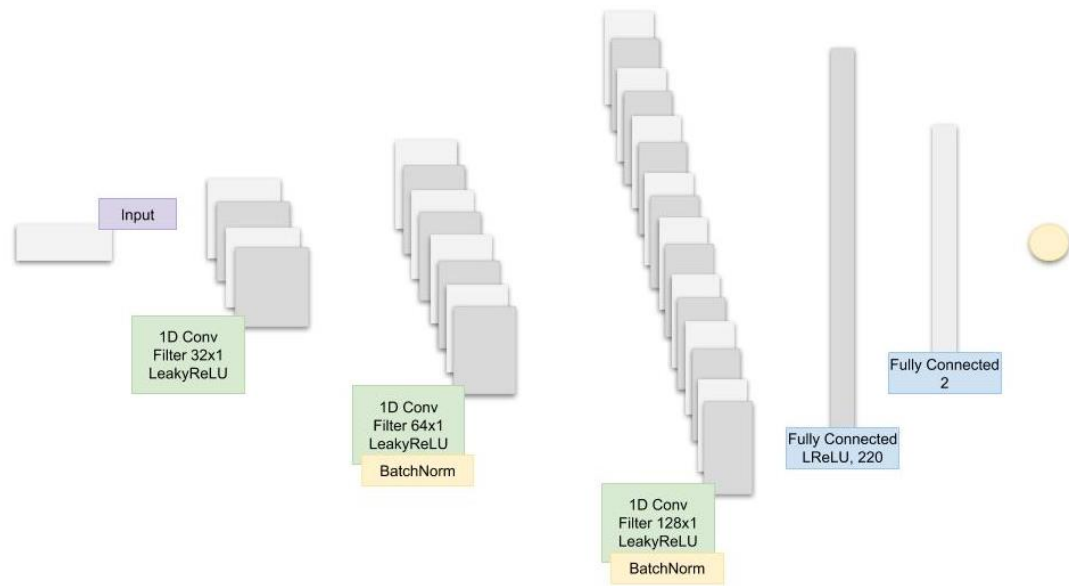


Рисунок 2.4 – Архітектура CNN

Нам потрібно зрозуміти, що впливає на те, чи буде ціна акцій рухатися вгору або вниз. Отже, потрібно включити якомога більше інформації. (Ми будемо використовувати щоденні дані – 1585 днів для навчання різних алгоритмів (70% даних, які у нас є) і прогнозувати наступні 680 днів (дані тесту). Потім ми порівняємо спрогнозовані результати з даними тесту (затримки). Кожен тип даних (ми будемо називати його функцією) більш детально пояснюється в наступних розділах, але в якості загального огляду ми будемо використовувати наступні функції:

Корельовані активи – це інші активи (будь-якого типу, необов'язково акції, такі як товари, валюта, індекси або навіть цінні папери з фіксованим доходом). Велика компанія, така як Goldman Sachs, очевидно, не «живе» в ізолюваному світі – вона залежить від багатьох зовнішніх факторів, включаючи їх конкурентів, клієнтів, глобальну економіку, геополітичну ситуацію, фіскальну політику і взаємодіє з ними. грошово-кредитна політика, доступ до капіталу і т. д.

Технічні індикатори – ми будемо включати найпопулярніші показники в якості незалежних функцій. Серед них – ковзне середнє за 7 і 21 день,

експоненціальне ковзне середнє, імпульс, смуги Боллінджера, MACD.

Фундаментальний аналіз – дуже важлива функція, що показує, чи може акція рухатися вгору або вниз. У фундаментальному аналізі можна використовувати дві функції:

1) аналіз продуктивності компанії з використанням звітів 10-K і 10-Q, аналіз ROE і P/E і т. д. (ми не будемо використовувати це);

2) новини – потенційно новини можуть вказувати на майбутні події, які потенційно можуть зрушити акції в певному напрямку. Ми будемо читати всі щоденні новини для Goldman Sachs і витягувати інформацію про те, чи є загальний настрій щодо Goldman Sachs в цей день позитивним, нейтральним або негативним (як оцінка від 0 до 1). Оскільки багато інвесторів уважно читають новини і приймають інвестиційні рішення, засновані (частково, звичайно) на новинах, існує досить висока ймовірність того, що, якщо, скажімо, новини для Goldman Sachs сьогодні надзвичайно позитивні, акції завтра виростуть. Одним з важливих моментів, ми будемо виконувати важливість функції (тобто наскільки вона показова для переміщення GS).

З метою створення точного прогнозу настроїв ми будемо використовувати Neural Language Processing (NLP). Ми будемо використовувати BERT – недавно анонсований Google підхід НЛП для трансферу навчання для отримання настроїв з фондових новин.

Перетворення Фур'є – поряд з денною ціною закриття ми створимо перетворення Фур'є, щоб узагальнити кілька довгострокових і короткострокових трендів. Використовуючи ці перетворення, ми усуваємо багато шуму (випадкових блукань) і створюємо наближення реального руху запасу. Наявність апроксимації трендів може допомогти мережі LSTM більш точно вибрати тенденції прогнозування.

Інтегрована модель авторегресії – ковзного середнього (ARIMA) – це був

один з найпопулярніших методів прогнозування майбутніх значень даних часових рядів (до нейронних мереж). Давайте додамо це і подивимось, чи буде це важливою прогнозуючою функцією.

Складені автоенкодери – більшість з вищезазначених функцій (фундаментальний аналіз, технічний аналіз і т. д.) були виявлені людьми після десятиліть досліджень. Можливо, існують приховані кореляції, які люди не можуть зрозуміти через величезної кількості точок даних, подій, активів, діаграм і т. д. За допомогою підсумовуваних автоенкодерів (типу нейронних мереж) ми, можливо, зможемо знайти нові типи функцій, які впливають на рух акцій та будемо використовувати їх в GAN.

Глибоке неконтрольоване навчання для виявлення аномалій в ціноутворенні опціонів. Ми будемо використовувати ще одну функцію – за кожен день ми будемо додавати ціну за опціон на 90 днів на акції Goldman Sachs. Опція ціноутворення сама по собі об'єднує безліч даних. Ціна опціонного контракту залежить від майбутньої вартості акцій (аналітики також намагаються передбачити ціну, щоб знайти найбільш точну ціну для опціону колл). Використовуючи глибоке неконтрольоване навчання (самоорганізуються карти), ми постараємось виявити аномалії в цінах кожного дня. Аномалія (наприклад, різка зміна цін) може вказувати на подію, яка може бути корисним для LSTM для вивчення загальної моделі акцій.

Далі, маючи так багато функцій, нам потрібно виконати декілька важливих кроків:

- Виконати статистичні перевірки «якості» даних.

Якщо дані, які ми створюємо, мають недоліки, то, якими б витонченими не були наші алгоритми, результати не будуть позитивними. Перевірки включають перевірку того, що дані не страждають від гетероскедастичності, мультиколінеарності або послідовної кореляції.

- Створити особливість важливості.

Якщо особливість (наприклад, інша акція або технічний індикатор) не має пояснювальної сили для акції, яку ми хочемо передбачити, тоді нам не потрібно використовувати її при навчанні нейронних мереж. Ми будемо використовувати XGBoost (eXtreme Gradient Boosting), тип алгоритмів прискореної регресії дерева.

На заключному етапі підготовки даних ми також створимо власні портфелі з використанням аналізу основних компонентів (PCA), щоб зменшити розмірність функцій, створених з автоенкодерів.

2.7 Навчання GAN

Оскільки GAN містить дві окремо навчені мережі, його алгоритм навчання має враховувати дві складності.

GAN повинні поєднувати два різних типи навчання (генератор і дискримінатор).

Конвергенцію GAN складно ідентифікувати.

У генератора і дискримінатора різні процеси навчання. Так як же навчити GAN в цілому?

Навчання GAN проходить в чергуючі періоди:

Дискримінатор тренується для однієї або декількох епох.

Генератор тренується для однієї або декількох епох.

Повторюємо кроки 1 і 2, щоб продовжити навчання ланцюгів генератора і дискримінатора.

Ми підтримуємо постійний генератор на етапі навчання дискримінатора. Оскільки навчання дискримінатора намагається зрозуміти, як відрізнити справжні дані від фальшивих, воно повинно навчитися розпізнавати недоліки генератора. Це інша проблема для ретельно навченого генератора, ніж для непідготовленого генератора, що видає випадковий результат.

Точно так само ми підтримуємо постійний дискримінатор під час фази навчання генератора. В іншому випадку генератор буде намагатися вразити рухому ціль і ніколи не зможе зійтися.

Саме ці дії дозволяють GAN вирішувати складні для генеративні проблеми. Ми приступаємо до вирішення складної проблеми генерації, почавши з набагато більш простої задачі класифікації. І навпаки, якщо ви не можете навчити класифікатор розрізняти реальні та згенеровані дані навіть для вихідних даних

генератора випадкових чисел, ви не зможете почати навчання GAN.

У міру того, як генератор вдосконалюється з навчанням, продуктивність дискримінатора погіршується, тому що дискримінатор не може легко відрізнити реальне від фальшивого. Якщо генератор працює ідеально, то точність дискримінатора становить 50%. Фактично, дискримінатор підкидає монетку, щоб зробити прогноз.

Цей прогрес створює проблему для конвергенції GAN в цілому: зворотний зв'язок дискримінатора згодом стає менш значущим. Якщо GAN продовжує навчання після того моменту, коли дискримінатор дає повністю випадковий зворотний зв'язок, тоді генератор починає навчатися на небажаному зворотному зв'язку, і його власна якість може погіршитися.

2.8 Висновки до розділу 2

Проведений аналіз показав, що використовувані на фондовому ринку прогностичні інструменти не мають прийнятної рівня прогностичної сили.

Така ситуація обумовлена і досить слабкою теоретичною основою прогнозування фінансових інструментів фондового ринку. Це залишає проблему розробки нового теоретико-методологічного підходу до прогнозування динаміки курсових вартостей акцій фондового ринку актуальною.

Найбільш перспективним напрямком створення більш ефективних прогностичних інструментів представляється ведення розробок в області ШІ.

Наявні розробки в цій області – нейромережі вже володіють деякими потенційними перевагами перед традиційними підходами та іншими математичними підходами. Наприклад, вони дозволяють враховувати найбільшу

кількість факторів, здатні вирішувати завдання, що не мають алгоритму та ін. Однак, нейромережі володіють і істотними недоліками, які практично зводять нанівець їх гідності, наприклад, відсутність безперервного перенавчання нейромережі в процесі виконання завдання досить швидко робить її неадекватною ситуації на ринку.

РОЗДІЛ 3 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА

3.1. Відповідні активи

Які активи можуть вплинути на рух акцій GS? Добре розуміння компанії, напрямків її діяльності, конкурентного середовища, залежностей, постачальників, типу клієнта тощо є дуже важливим для вибору правильного набору відповідних активів:

1. Першими є компанії, подібні до GS. Ми додамо JPMorgan Chase та Morgan Stanley, серед іншого, до набору даних.
2. Як інвестиційний банк, Goldman Sachs залежить від світової економіки. Погана або нестабільна економіка означає відсутність злиття та поглинання або випуску акцій, і, можливо, обмежений прибуток від власних торгів. Тому ми будемо включати показники світової економіки. Крім того, ми включимо курс LIBOR (у доларах США та у фунтах стерлінгів), оскільки аналітики можуть визначити шоки в економіці для встановлення цих ставок та інші цінні папери FI .
3. Щоденний індекс волатильності (VIX) – з причини, описаної в попередньому пункті.
4. Складені індекси – такі як індекси NASDAQ та NYSE (із США), FTSE100 (Великобританія), Nikkei225 (Японія), Hang Seng та BSE Sensex (APAC).
5. Валюти – глобальна торгівля багато разів відображається на тому, як рухаються валюти, тому ми будемо використовувати кошик валют (таких як USDJPY, GBPUSD тощо) як особливості.

3.2. Технічні показники

Ми вже розглянули, що таке технічні показники і чому ми їх використовуємо, тому давайте перейдемо прямо до коду. Ми створимо технічні показники лише для GS, зображено на рисунках 3.1 і 3.2.

ціна	ma7	ma21	26ема	12ема	MACD
153.130005	152,374285	164.220476	160,321839	156,655072	-3,666767
156,940002	152,777143	163,653809	160,014868	156,700048	-3,314821
157,229996	153.098572	162,899047	159,766235	156,783365	-2,982871
150,679993	153.069999	161,686666	158,967168	155,827031	-3,140137
154,160004	153,449999	160,729523	158,550196	155,566566	-2,983631

Рисунок 3.1 – Технічні показники, частина 1

20сд	верхній_діапазон	нижня_діапазон	ема	імпульс	log_momentum
9,607375	183,435226	145,005726	152,113609	152,130005	5.024735
9.480630	182,615070	144,692549	155,331205	155,940002	5.049471
9.053702	181,006450	144,791644	156,597065	156,229996	5.051329
8,940246	179,567157	143.806174	152,652350	149,679993	5,008500
8.151912	177,033348	144,425699	153,657453	153,160004	5.031483

Рисунок 3.2 – Технічні показники, частина 2

Отже, ми маємо технічні показники (включаючи MACD, смуги Боллінджера тощо) для кожного торгового дня. Всього ми маємо 12 технічних показників.

Давайте візуалізуємо останні 400 днів цих показників на рисунку 3.3.

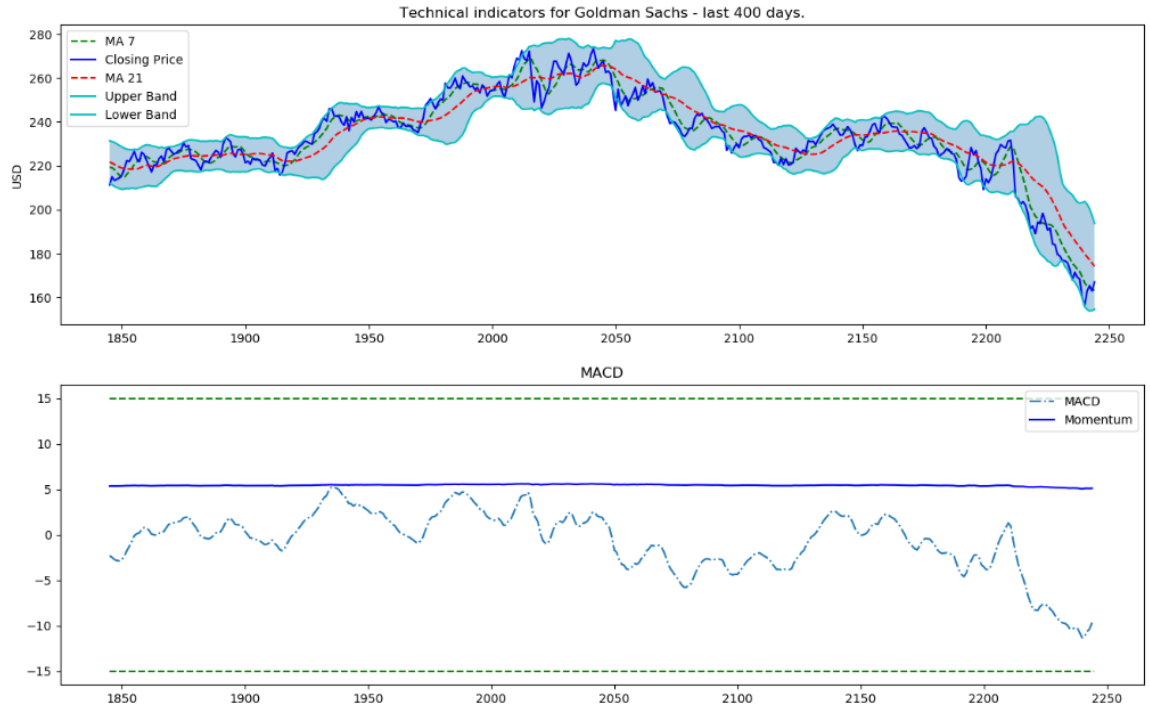


Рисунок 3.3 – останні 400 днів технічних показників

3.3. Фундаментальний аналіз

Для фундаментального аналізу ми проведемо аналіз настроїв щодо всіх щоденних новин про GS. Використовуючи сигмоїд в кінці, результат буде від 0 до 1. Чим ближче оцінка до 0 – тим більше негативних новин (ближче до 1 означає позитивні настрої). Для кожного дня ми створимо середній щоденний бал (як число від 0 до 1) і додамо його як функцію.

3.3.1. Представлення двостороннього вбудування від трансформаторів – BERT

З метою класифікації новин як позитивних чи негативних (або нейтральних) ми будемо використовувати BERT, який є попередньо підготовленим мовним поданням.

Попередньо оброблені моделі BERT вже доступні в MXNet / Gluon. Нам просто потрібно створити їх і створити два (довільні числа) Denseшари, переходячи до softmax - оцінка становить від 0 до 1.

3.4. Перетворення Фур'є для аналізу тенденцій

Перетворення Фур'є беруть функцію і створюють ряд синусоїд (з різними амплітудами та кадрами). У поєднанні ці синусоїди відповідають початковій функції. Математично кажучи, перетворення виглядають так:

$$G(f) = \int_{-\infty}^{\infty} g(t) e^{-i 2 \pi f t} dt$$

Ми використовуватимемо перетворення Фур'є, щоб виділити глобальні та місцеві тенденції ціни GS, а також трохи дезактивувати його. Тож давайте подивимось, як це працює на рисунку 3.4.

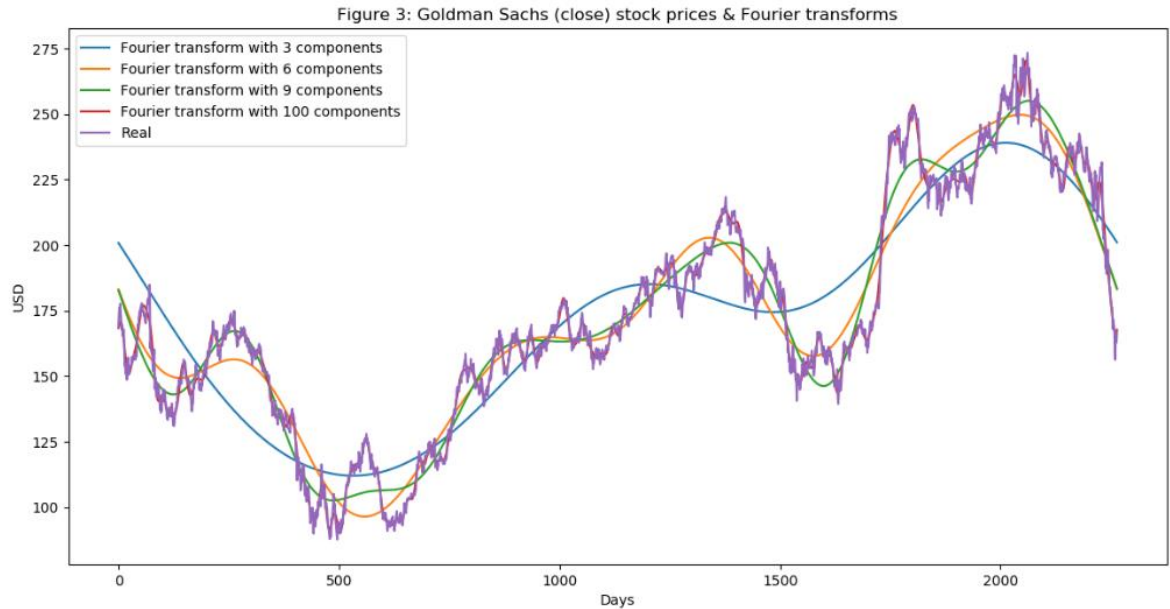


Рисунок 3.4 – Ціна акцій і перетворення Фур'є

Як бачимо на рисунку 3.4, чим більше компонентів з перетворення Фур'є ми використовуємо, тим ближче функція наближення до реальної ціни акцій (перетворення 100 компонентів майже ідентична вихідній функції – червона та фіолетова лінії майже перекриваються). Ми використовуємо перетворення Фур'є з метою вилучення довгострокових та короткострокових тенденцій, тому будемо використовувати перетворення з 3, 6 та 9 компонентами.

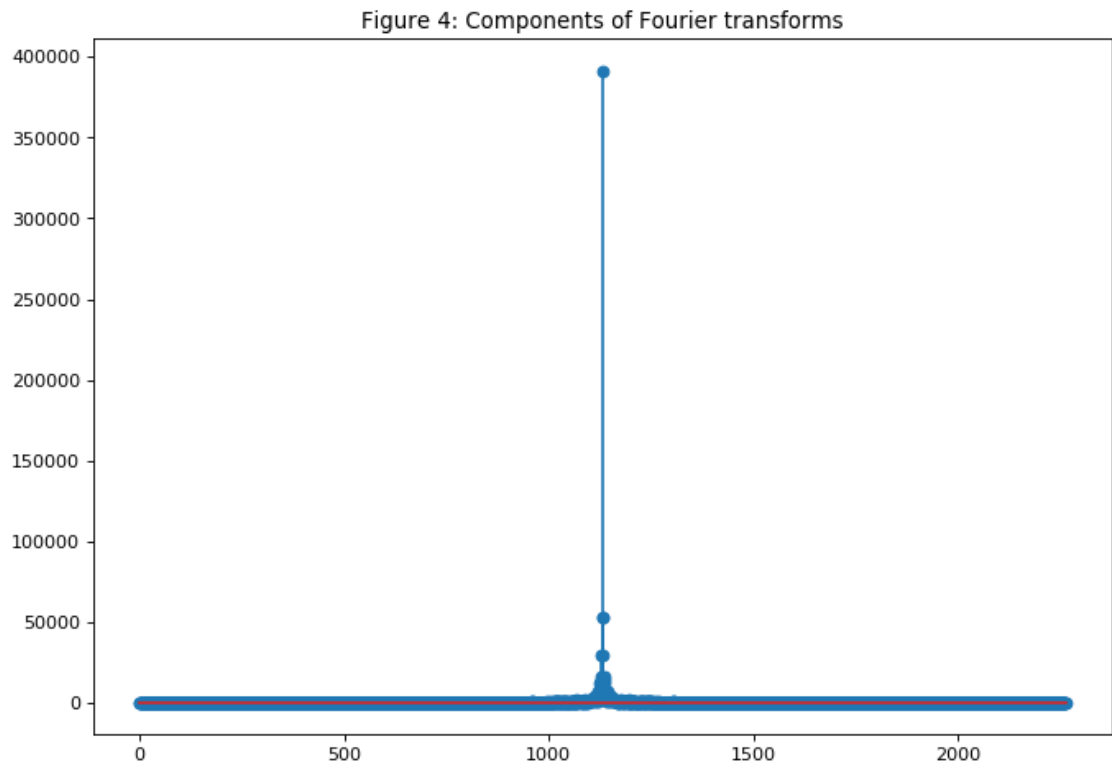


Рисунок 3.5 – Компоненти перетворення Фур'є

З рисунка 3.5 можна зробити висновок, що перетворення з 3 компонентами служить довгостроковою тенденцією.

Інший прийом, який використовується для відключення даних, - це вейвлети викликів. Вейвлети та перетворення Фур'є дали подібні результати, тому ми будемо використовувати лише перетворення Фур'є.

3.5. ARIMA як особливість

ARIMA – це метод прогнозування даних часових рядів. ARIMA не буде нашим остаточним прогнозом, ми будемо використовувати його як техніку, щоб трохи дезонізувати ціну акції і (можливо) витягти деякі нові особливості.

Виведемо на екран результати моделі ARIMA, рисунок 3.5 та побудуємо графік похибок, рисунок 3.6.

ARIMA Model Results						
=====						
Dep. Variable:	D.GS	No. Observations:	2264			
Model:	ARIMA(5, 1, 0)	Log Likelihood	-5465.888			
Method:	css-mle	S.D. of innovations	2.706			
Date:	Wed, 09 Jan 2019	AIC	10945.777			
Time:	10:28:07	BIC	10985.851			
Sample:	1	HQIC	10960.399			
=====						
	coef	std err	z	P> z	[0.025	0.975]

const	-0.0011	0.054	-0.020	0.984	-0.106	0.104
ar.L1.D.GS	-0.0205	0.021	-0.974	0.330	-0.062	0.021
ar.L2.D.GS	0.0140	0.021	0.665	0.506	-0.027	0.055
ar.L3.D.GS	-0.0030	0.021	-0.141	0.888	-0.044	0.038
ar.L4.D.GS	0.0026	0.021	0.122	0.903	-0.039	0.044
ar.L5.D.GS	-0.0522	0.021	-2.479	0.013	-0.093	-0.011
Roots						
=====						
	Real	Imaginary	Modulus	Frequency		

AR.1	-1.7595	-0.0000j	1.7595	-0.5000		
AR.2	-0.5700	-1.7248j	1.8165	-0.3008		
AR.3	-0.5700	+1.7248j	1.8165	0.3008		
AR.4	1.4743	-1.0616j	1.8168	-0.0993		
AR.5	1.4743	+1.0616j	1.8168	0.0993		

Рисунок 3.6 – Модель ARIMA

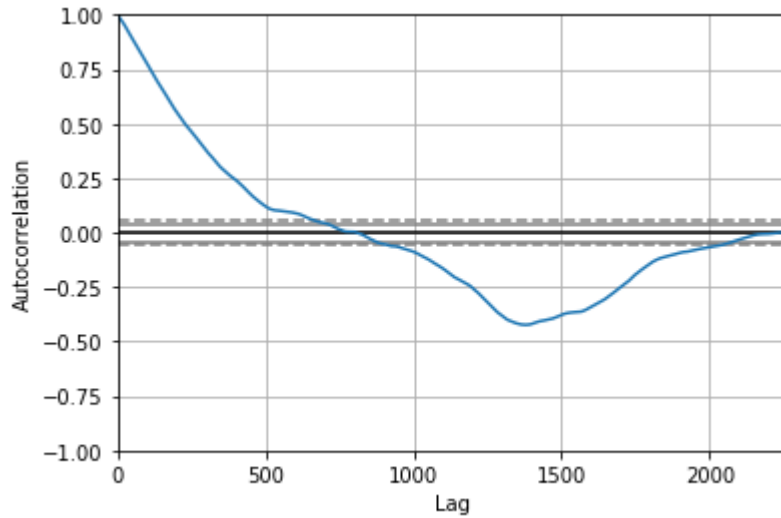


Рисунок 3.7 – Похибка

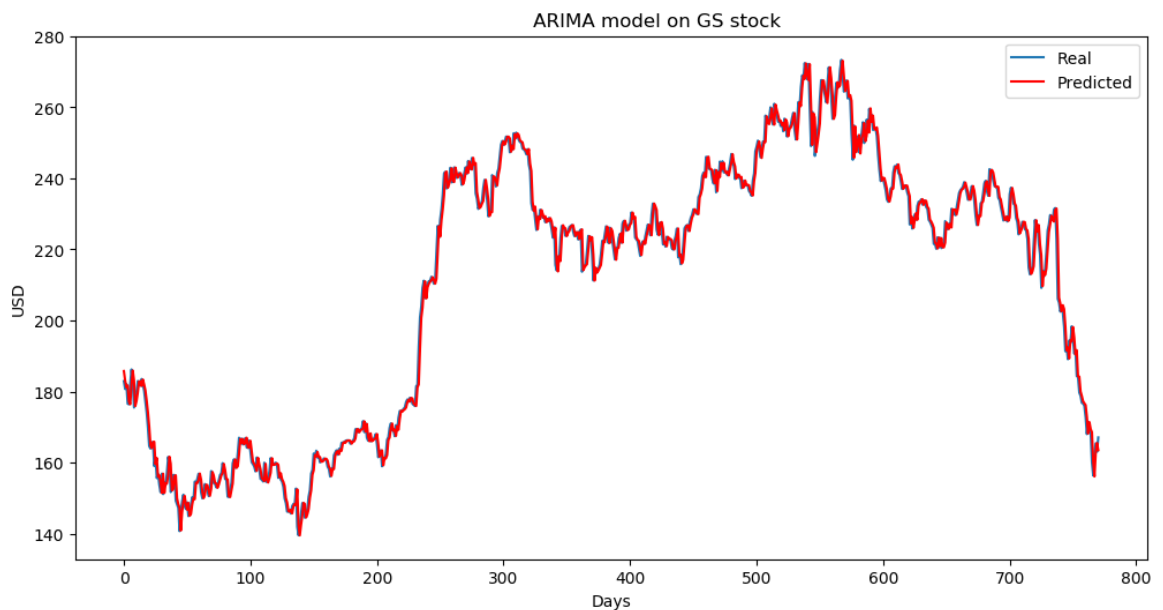


Рисунок 3.8 – ARIMA для GS

Як ми бачимо на рисунку 3.8, ARIMA дає дуже хорошу апроксимацію реальної ціни акцій. Ми будемо використовувати прогнозовану ціну через ARIMA як вхідну функцію до LSTM, оскільки, як ми вже згадували раніше, ми хочемо охопити якомога більше можливостей та моделей Goldman Sachs. Ми проходимо тестування MSE (середня квадратична помилка) 10,151, що саме по

собі не є поганим результатом (враховуючи, що ми маємо багато даних тесту), але все одно ми будемо використовувати його лише як функцію в LSTM.

3.6. Статистичні перевірки

Для вихідних моделей дуже важливо забезпечити якість даних. Для того, щоб переконатися, що наші дані придатні, ми виконаємо пару простих перевірок, щоб переконатися, що результати, яких ми досягаємо і спостерігаємо, справді є реальними, а не скомпрометованими через те, що основний розподіл даних страждає від фундаментальних помилок.

3.6.1. Гетероскедастичність, мультиколінеарність, послідовна кореляція

1. Умовна гетероскедастичність виникає, коли умови помилки (різниця між передбачуваним значенням регресії та реальним значенням) залежать від даних – наприклад, умови помилки зростають, коли точка даних (вздовж осі x) зростає.
2. Мультиколінеарність – це коли терміни помилок (їх також називають залишками) залежать один від одного.
3. Послідовна кореляція – це коли одні дані (ознака) є формулою (або повністю залежать) від іншої ознаки.

3.7. Функціональна інженерія

Отже, після додавання всіх типів даних (відповідних активів, технічних показників, фундаментального аналізу, Фур'є та Аріми) ми маємо загалом 112 функцій за 2265 днів (як уже згадувалося раніше, однак лише 1585 днів стосуються навчальних даних) .

Ми також матимемо ще деякі функції, згенеровані з автокодерів.

3.7.1.Важливість функції з XGBoost

Маючи стільки функцій, ми повинні врахувати, чи всі вони насправді вказують на напрямок, який буде здійснювати запас GS. Наприклад, ми включили в набір даних ставки LIBOR, деноміновані в доларах США, оскільки ми вважаємо, що зміни в LIBOR можуть свідчити про зміни в економіці, а це, в свою чергу, може свідчити про зміни в поведінці акцій GS. Але нам потрібно перевірити. Існує багато способів перевірити важливість особливостей, але той, який ми застосуємо, використовує XGBoost, оскільки він дає один із найкращих результатів як для класифікації, так і для проблем регресії.

Оскільки набір функцій є досить великим, для представлення тут ми будемо використовувати лише технічні показники. Під час тестування важливості реальних характеристик усі вибрані функції виявились дещо важливими, тому ми не будемо нічого виключати під час навчання GAN.

Давайте побудуємо помилки навчання на рисунку 3.9 та перевірки на рисунку 3.10 , щоб спостерігати за тренуванням та перевіряти переобладнання.



Рисунок 3.9 – Помилки навчання та перевірки

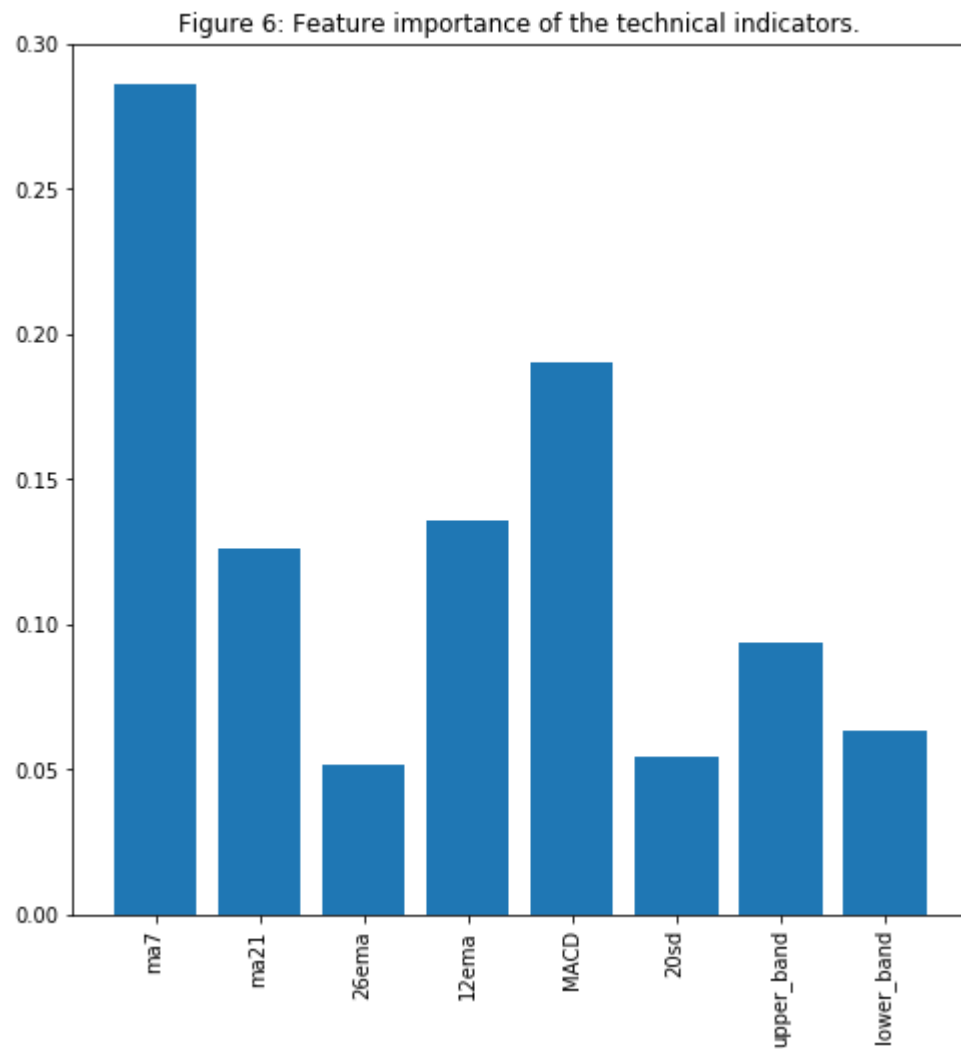


Рисунок 3.10 – Важливі технічні індикатори

Важливі технічні індикатори зображено на рисунку 3.10. Не дивно (для тих, хто має досвід у біржовій торгівлі), що MA7, MACD та BB є одними з важливих особливостей.

3.8. Видобування високорівневих функцій за допомогою автоматичних кодерів з накопиченням

Перш ніж перейти до автокодерів, ми розглянемо альтернативну функцію активації.

3.8.1. Функція активації – GELU (Гауссова помилка)

GELU нещодавно було запропоновано лінійні об'єднання Гауссових помилок. У статті автори показують кілька випадків, коли нейронні мережі, що використовують GELU, перевершують мережі, використовуючи ReLU як активацію.

Ми використовуватимемо GELU для автокодерів.

Нижче показано логіку, що лежить в основі математики GELU. Це не фактична реалізація як функція активації. Мені довелося впровадити GELU всередині MXNet.

Давайте візуалізуємо GELU, ReLU і LeakyReLU(останній в основному використовується в GAN – ми також його використовуємо) на рисунках 3.11 та 3.12.

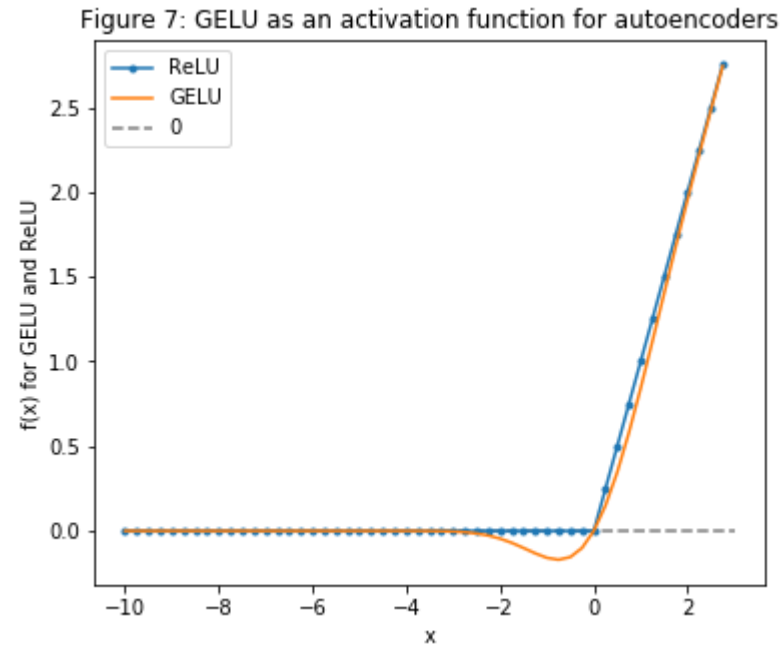


Рисунок 3.11 – GELU як функція активації автокодерів

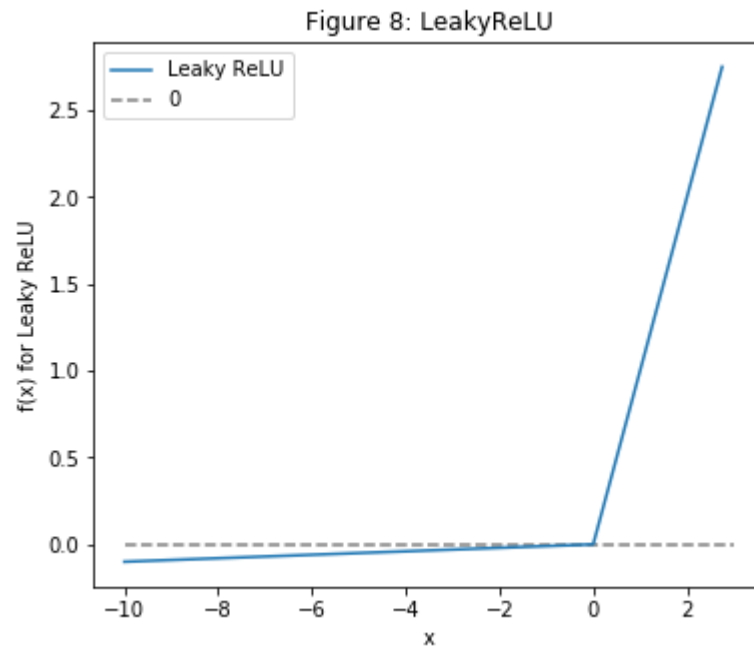


Рисунок 3.12 – LeakyReLU

Гаразд, повернемось до автокодерів, зображених нижче (зображення лише схематично, воно не відображає реальну кількість шарів, одиниць тощо)

Зазвичай в автокодерах кількість кодерів = кількість декодерів. Однак ми хочемо витягти функції вищого рівня (а не створювати той самий вхід), щоб ми могли пропустити останній шар у декодері. Ми досягаємо цього, створюючи кодер і декодер з однаковою кількістю шарів під час навчання (зображено на рисунку 3.13), але коли ми створюємо вихідні дані, ми використовуємо шар поруч із єдиним, оскільки він містив би функції вищого рівня.

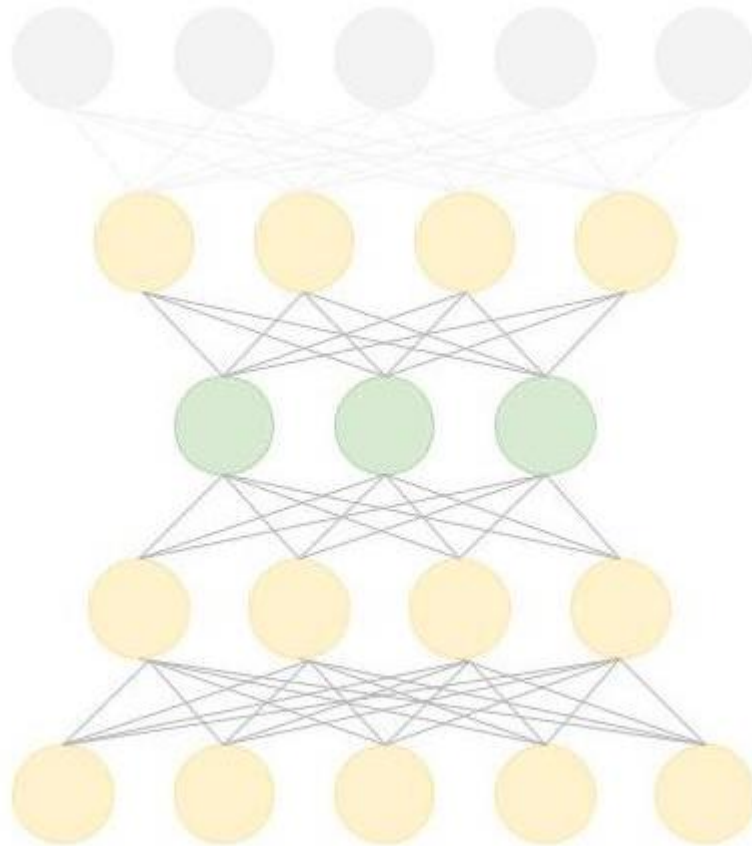


Рисунок 3.13 – Автокодери

Виведемо на екран кількість шарів в кодері та декодері, рисунок 3.14.

```
VAE(
  (encoder): HybridSequential(
    (0): Dense(None -> 400, Activation(relu))
    (1): Dense(None -> 400, Activation(relu))
    (2): Dense(None -> 400, Activation(relu))
    (3): Dense(None -> 4, linear)
  )
  (decoder): HybridSequential(
    (0): Dense(None -> 400, Activation(relu))
    (1): Dense(None -> 400, Activation(relu))
    (2): Dense(None -> 400, Activation(relu))
    (3): Dense(None -> 11, Activation(sigmoid))
  )
)
```

Рисунок 3.14 – кількість шарів в кодері та декодері

Отже, з рисунка 3.14 ми маємо 3 шари (з 400 нейронами в кожному) як в кодері, так і в декодері.

Ми створили ще 112 функцій з автокодера. Оскільки ми хочемо мати лише функції високого рівня (загальні схеми), ми створимо портфель Eigen на новостворених 112 об'єктах за допомогою аналізу основних компонентів (PCA). Це зменшить розмірність (кількість стовпців) даних. Описова здатність портфеля Eigen буде такою ж, як і у вихідних 112 функцій.

3.8.2. Eigen портфоліо з PCA

Отже, для пояснення 80% дисперсії нам потрібні 84 (із 112) особливостей. Це ще багато. Отже, наразі ми не будемо включати створені автокодером функції. Будемо працювати над створенням архітектури автокодера, в якій ми отримуємо вихідні дані з проміжного шару (не останнього) і підключаємо його до іншого

Dense шару, скажімо, з 30 нейронами. Таким чином, ми:

- 1) будемо витягувати лише функції вищого рівня;
- 2) матимемо значно меншу кількість стовпців.

Архітектура LSTM дуже проста – один LSTM шар із 112 вхідними одиницями (оскільки у нас є 112 функцій у наборі даних) і 500 прихованих одиниць, і один Dense шар з 1 висновком – ціна на кожен день. Ініціатором є Xavier, і ми будемо використовувати втрату L1 (що є середньою абсолютною втратою помилок при регуляризації L1).

Ми використаємо 500 нейронів у шарі LSTM та використаємо ініціалізацію Xavier. Для регуляризації будемо використовувати L1.

Як ми бачимо, вхідними даними LSTM є 112 функцій (`dataset_total_df.shape[1]`), які потім переходять у 500 нейронів на рівні LSTM, а потім трансформуються в один вихід – значення ціни акцій.

Логіка LSTM полягає в тому, що ми беремо дані за 17 (`sequence_length`) днів (знову ж таки, дані є ціною акцій на акції GS щодня + всі інші функції цього дня – відповідні активи, настрої тощо) і намагаємося передбачити 18-й день.

3.8.3.Планувальник швидкості навчання

Одним з найважливіших гіперпараметрів є швидкість навчання. Встановлення швидкості навчання майже для кожного оптимізатора (наприклад, SGD, Адама або RMSProp) має вирішальне значення при навчанні нейронних мереж, оскільки воно контролює як швидкість конвергенції, так і кінцеву продуктивність мережі. Однією з найпростіших стратегій швидкості навчання є наявність фіксованої швидкості навчання протягом усього навчального процесу. Вибір невеликої

швидкості навчання дозволяє оптимізатору знайти хороші рішення, але це відбувається за рахунок обмеження початкової швидкості конвергенції. Зміна швидкості навчання з часом може подолати цю компромісну ситуацію. На рисунку 3.15 можна побачити швидкість навчання для кожної епохи.

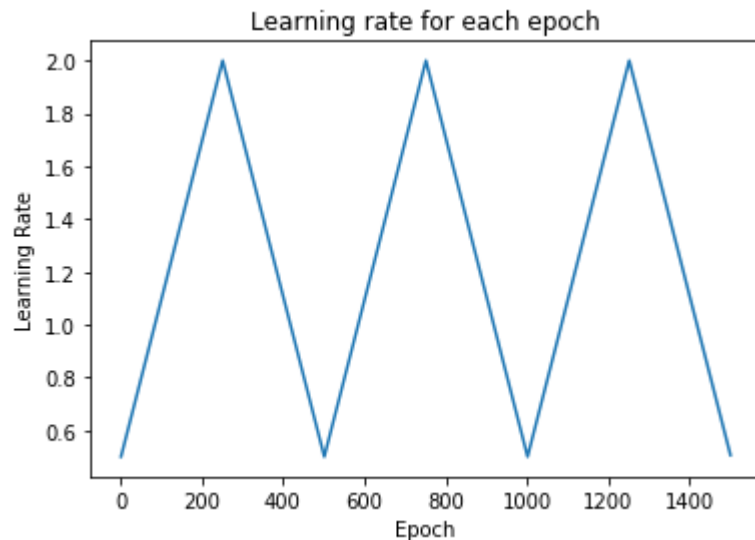


Рисунок 3.15 – Швидкість навчання для кожної епохи

3.8.4. Як запобігти переобладнанню та компромісу між зміщенням

Маючи безліч функцій та нейронних мереж, нам потрібно переконатись, що ми запобігаємо переобладнанню та пам'ятаємо про загальну ціну.

Ми використовуємо кілька методів для запобігання переобладнанню (не тільки в LSTM, але і в CNN та в автоматичних кодерах):

- Забезпечення якості даних. Ми вже проводили статистичні перевірки та переконувались, що дані не страждають від мультиколінеарності чи послідовної автокореляції. Далі ми виконали перевірку важливості функції для кожної функції. Нарешті, первинний вибір ознак (наприклад, вибір відповідних активів,

технічних показників тощо) був зроблений з певними знаннями в галузі про механіку, що лежить в основі роботи фондових ринків.

- Регулювання (або покарання ваг). Дві найбільш широко використовувані методи регуляризації - LASSO (L1) та Ридж (L2)). L1 додає середню абсолютну помилку, а L2 додає середню квадратичну помилку до втрати. Не вдаючись у занадто багато математичних деталей, основні відмінності полягають у тому, що лассо-регресія (L1) виконує як вибір змінної, так і усадку параметрів, тоді як регресія Риджа виконує лише усадку параметрів і в підсумку включає всі коефіцієнти в модель. За наявності корельованих змінних регресія хребта може бути кращим вибором. Крім того, регресія хребта найкраще працює в ситуаціях, коли найменш квадратні оцінки мають більшу дисперсію. Отже, це залежить від мети нашої моделі. Вплив двох типів регуляризацій досить різний. Хоча вони обидва карають великі ваги, регуляризація L1 призводить до недиференційованої функції при нулі. Регуляризація L2 надає перевагу меншим вагам, але регуляризація L2 надає перевагу вагам, які дорівнюють нулю. Тому, з регуляризацією L1 ми можемо отримати розріджену модель – таку з меншою кількістю параметрів. В обох випадках параметри регульованих моделей L1 та L2 «стискаються», але у випадку регуляризації L1 усадка безпосередньо впливає на складність (кількість параметрів) моделі. Точно, регресія хребта найкраще працює в ситуаціях, коли найменш квадратні оцінки мають більшу дисперсію. L1 є надійнішим у порівнянні з випадковими, використовується, коли даних не вистачає, і створює важливість особливостей. Ми будемо використовувати L1. Регресія хребта найкраще працює в ситуаціях, коли найменш квадратні оцінки мають більшу дисперсію. L1 є надійнішим у порівнянні з випадковими, використовується, коли даних не вистачає, і створює важливість особливостей.

- Вибування. Випадаючі шари випадковим чином видаляють вузли у прихованих шарах.

- Рання зупинка. Іншим важливим фактором при побудові складних нейронних мереж є компроміс між зміщенням і дисперсією. В основному, помилка, яку ми отримуємо при навчанні мереж, є функцією упередженості, дисперсії та незвідимої помилки – σ (помилка через шум та випадковість). Найпростіша формула компромісу:

$$MSE = \text{bias}^2 + \text{variance} + \sigma^2$$

- Упередженість. Упередженість вимірює, наскільки добре навчений (на базі даних навчального набору) алгоритм може узагальнити на невидимі дані. Висока упередженість (недостатнє оснащення) означає, що модель не може добре працювати з невидимими даними.

- Дисперсія. Дисперсія вимірює чутливість моделі до змін у наборі даних. Велика дисперсія полягає в переобладнанні.

3.9 Гіперпараметри

Гіперпараметри, які ми будемо відстежувати та оптимізувати:

- 1) `batch_size`: розмір партії LSTM та CNN;
- 2) `cnn_lr`: швидкість навчання CNN;
- 3) `strides`: кількість кроків у CNN;
- 4) `lrelu_alpha`: альфа для LeakyReLU у GAN;
- 5) `batchnorm_momentum`: імпульс для нормалізації партії в CNN;
- 6) `padding`: заповнення в CNN;
- 7) `kernel_size`: розмір ядра в CNN;
- 8) `dropout`: відсівання в LSTM;
- 9) `filters`: початкова кількість фільтрів.

Ми будемо тренувати понад 200 epochs.

Після того, як GAN тренується протягом 200 епох, він буде реєструвати MAE (що є функцією помилки в LSTM, \$ G \$) і передавати його як значення винагороди навчанню з підсилення, яке вирішить, чи слід змінювати гіперпараметри продовження навчання з однаковим набором гіперпараметрів. Як було описано далі, цей підхід призначений виключно для експериментів з RL.

Якщо RL вирішить, що оновить гіперпараметри, він викличе байєсівську оптимізацію (обговорену нижче), яка дасть наступний найкращий набір гіперпарам.

3.10. Підсилення навчання для оптимізації гіперпараметрів

Чому ми використовуємо навчання підкріплення при оптимізації гіперпараметрів? Фондові ринки весь час змінюються. Навіть якщо нам вдається навчити наш GAN та LSTM створювати надзвичайно точні результати, результати можуть бути дійсними лише протягом певного періоду. Це означає, що нам потрібно постійно оптимізувати весь процес. Для оптимізації процесу ми можемо:

- Додавати або видаляти функцій (наприклад, додавання нових акцій або валют, які можуть корелювати).
- Удосконалювати наші моделі глибокого навчання. Одним з найважливіших способів вдосконалення моделей є гіперпараметри. Знайшовши певний набір гіперпараметрів, нам потрібно вирішити, коли їх змінювати, а коли використовувати вже відомий набір (розвідка проти експлуатації). Крім

того, ринок акцій являє собою безперервний простір, який залежить від мільйонів параметрів.

3.11 Результати роботи

Результати прогнозування зображені на рисунках 3.16-3.18

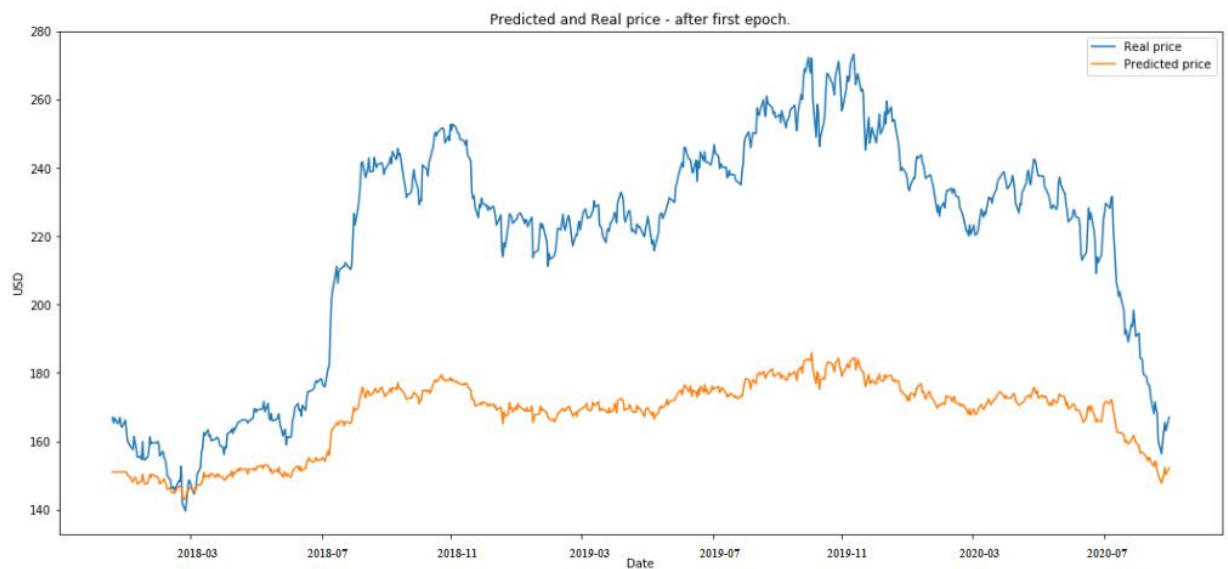


Рисунок 3.16 – Реальна і прогнозована ціна після першої епохи



Рисунок 3.17 – Реальна і прогнозована ціна після перших 50-ти епох

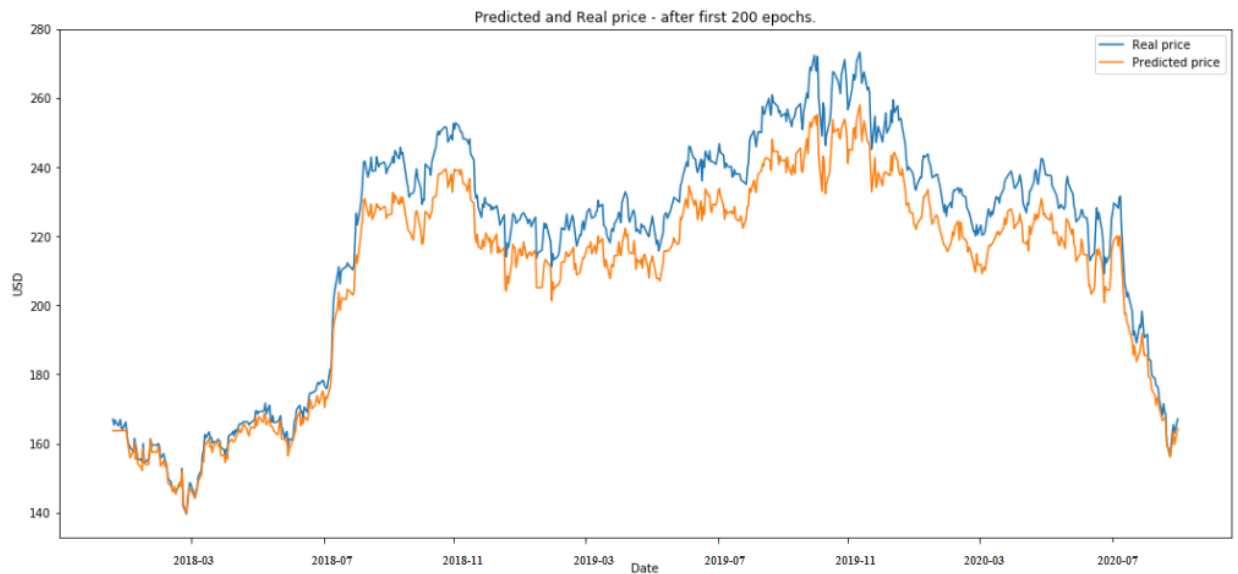


Рисунок 3.18 – Реальна і прогнозована ціна після перших 200-та епох

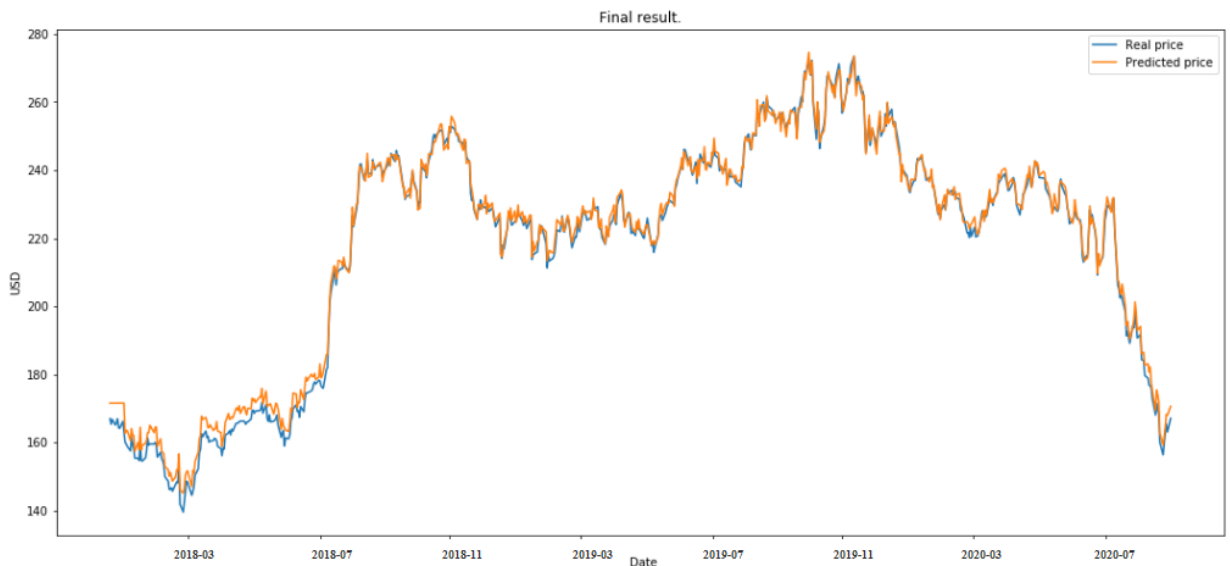


Рисунок 3.19 – Реальна і прогнозована ціна, фінальний результат

Як бачимо з рисунка 3.19 нам вдалося отримати досить точний прогноз, наближений до реального. Точність прогнозу нашої моделі складає 81,14%, тоді як точність прогнозу ARIMA в класичному вигляді 59,67%, GAN – 73,65%, LSTM – 74,18%, глибока LSTM – 63,78%. Отже, вдалося створити модель

прогнозування, яка дає більш точний прогноз в порівнянні з існуючими моделями.

3.12 Висновки до розділу 3

Точне прогнозування фондових ринків є складним завданням, оскільки існують мільйони подій та передумов для того, щоб акціонерні товариства рухались у певному напрямку. У цій роботі було запропоновано просту у використанні модель прогнозування фондового, щоб допомогти дедалі більшій кількості нефінансових професійних звичайних інвесторів приймати рішення.

Було використано навчання підкріплення при оптимізації гіперпараметрів, адже фондові ринки весь час змінюються. Навіть якщо вдається навчити GAN та LSTM створювати надзвичайно точні результати, результати можуть бути дійсними лише протягом певного періоду. Це означає, що потрібно постійно оптимізувати весь процес. Заснована на мережі глибокого навчання, ця модель досягає здатності прогнозування, що перевершує інші базові методи, за допомогою змагальних тренувань, мінімізації втрат прогнозування напрямку та втрати прогнозованих помилок. Більше того, аналізуються ефекти циклів оновлення моделі на можливість прогнозування, а експериментальні результати показують, що менший цикл оновлення моделі може отримати кращі результати прогнозування. У майбутньому ми спробуємо створити середовище RL для тестування торгових алгоритмів, які вирішують, коли і як торгувати. Вихід з GAN буде одним із параметрів у середовищі.

РОЗДІЛ 4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

4.1 Опис ідеї проекту

Розділ описує розробку стартап-проекту на тему “Генеративно-змагальна нейронна мережа для прогнозування фондового ринку”.

Метою розділу є формування інноваційного мислення, підприємницького духу та формування здатностей щодо оцінювання ринкових перспектив і можливостей комерціалізації основних науково-технічних розробок, сформованих у попередній частині магістерської дисертації у вигляді розроблення концепції стартап-проекту в умовах висококонкурентної ринкової економіки глобалізаційних процесів.

Опис ідеї стартап-проекту Генеративно-змагальна нейронна мережа для прогнозування фондового ринку наведено у Таблиці 4.1.

Таблиця 4.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Надавання послуг прогнозування часових рядів	Використання для прогнозування курсів акцій	Можливість прогнозувати значення курсу акції
	Використання у проектах, в яких використовується прогнозування даних	Додатковий метод прогнозування даних персональних даних в докер-контейнерах

Аналіз потенційних техніко-економічних переваг ідеї порівняно із

пропозиціями конкурентів передбачає:

- визначення переліку техніко-економічних властивостей та характеристик ідеї;
- визначення попереднього кола конкурентів, проектів-конкурентів, товарів-замінників чи товарів-аналогів, що вже існують на ринку;
- збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів.

Відповідно до визначеного вище переліку проводиться порівняльний аналіз показників: для власної ідеї визначаються показники, що мають:

- гірші значення (W, слабкі);
- аналогічні (N, нейтральні) значення;
- кращі значення (S, сильні).

Визначення сильних, слабких та нейтральних характеристик ідеї стартап-проекту “Мікросервісний підхід до розробки клієнтської частини веб-застосувань” наведено у Таблиці 4.2.

Основними позитивними характеристиками товару є його низька ціна, індивідуальний підхід до кожної задачі, зручність у використанні

Метою розділу є формування інноваційного мислення, підприємницького духу та формування здатностей щодо оцінювання ринкових перспектив і можливостей комерціалізації основних науково-технічних розробок, сформованих у попередній частині магістерської дисертації у вигляді розроблення концепції стартап-проекту в умовах висококонкурентної ринкової економіки глобалізаційних процесів.

Таблиця 4.2 – Визначення сильних, слабких та нейтральних характеристик ідеї стартап-проекту

№ п/ п	Техніко- економічні характеристики ідеї	(Потенційні) товари/концепції конкурентів			W	N	S
		Мій проект	Novo Forecast	STATISTIC A			
1.	Форма виконання	Надавання послуг	Надавання послуг	Надавання послуг		+	
2.	Собівартість	Низька	Висока	Середня			+
3.	Наявність індивідуального підходу до кожної задачі	Є	Є	Нема		+	
4.	Функціонал	Середній	Дуже великий	Середній	+		
5.	Швидкість виконання	Середня	Середня	Низька		+	
	Зручність використання	Висока	Висока	Низька		+	

4.2 Технологічний аудит проекту

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології реалізації	Наявність Технології	Доступність технології
1	Створення програмної частини	Python	Наявна	Доступна, безкоштовна, легка у реалізації
		Python	Ненаявна	Доступна, безкоштовна, Середня важкість реалізації
	Зв'язок із користувачем	Текстовий файл	Наявна	Легкість використання як частини більш високорівневих проектів
		Console output	Наявна	Легкість читання користувачем

З таблиці 4.3 можна побачити, що всі технології для реалізації мого проекту є доступними і безкоштовними, проте для створення програмного продукту було обрано мову Python.

4.3 Аналіз ринкових можливостей

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть зашкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту з урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Для того щоб спланувати напрямки розвитку проекту необхідно визначити ринкові можливості та ринкові загрози. Також необхідно визначити потреби потенційних клієнтів та пропозицій конкурентів. Попередня характеристика ринку наведена у таблиці 4.4.

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап-проекту

Показники стану ринку (найменування)	Характеристика
Кількість головних гравців, од	100
Загальний обсяг продажу, грн/ум.од	25000 грн/у.о
Динаміка ринку (якісна оцінка)	Зростає
Наявність обмежень для входу (вказати характер обмежень)	Відсутні
Специфічні вимоги до стандартизації та сертифікації	Відсутні
Середня норма рентабельності в галузі (або по ринку), %	10%

Було проаналізовано обсяг ринку та динаміку його розвитку, а також

наявність попиту. За результатами аналізу можна сказати, що обмежень для входу на ринок немає, динаміка ринку зростає, ринок є рентабельним.

Для визначення ключових елементів реалізації проекту, необхідно чітко визначити цільову аудиторію, особливості поведінки цільових груп та вимоги, що користувачі висувають до продукту. У таблиці 4.5 наведена характеристика потенційних клієнтів.

Таблиця 4.5 - Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Точне та швидке прогнозування різних даних.	Аудиторія: клієнти, у яких є потреба прогнозувати дані. Сегменти: індивідуальні користувачі, підприємства будь-якого розміру.	Для сегменту дрібних користувачів більш характерні програми, в якій використовується 1-2 методи прогнозування Підприємства зацікавлені у великих інфраструктурах програм щоб забезпечити надійність.	Усім споживачам важливі надійність та невиский час прогнозуван ня.

Виходячи з проведеного аналізу можна сказати, що для охоплення всіх потенційних споживачів, необхідно забезпечити них хорошим зворотнім зв'язком, та створити зручне рішення для задоволення їх потреб.

Ринкові загрози – події, настання яких може несприятливо вплинути на підприємство.

Для вдалого майбутнього проекту необхідно врахувати ситуації, що можуть виникнути в майбутньому та бути готовими до активних дій у разі їх появи. Аналіз загроз наведений у таблиці 4.6.

Таблиця 4.6 - Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Зростаюча конкуренція.	Зі зростом популярності зросла і пропозиція програм, які прогнозують дані.	Покращити функціональність програми.
2	Зміна потреб користувачів.	Користувачам необхідний сервіс з більшим/новим функціоналом.	Розроблення гнучкої архітектури програмного забезпечення для легшого впровадження нового функціоналу.
3	Економічні чинники.	Зменшення купівельної здатності цільової аудиторії.	Здешевлення продукту.

Ринкові можливості – це сприятливі обставини, які підприємство може використовувати для отримання переваг. Слід зазначити, що можливостями з погляду SWOT-аналізу є не всі можливості, які існують на ринку, а тільки ті, які

можна використовувати.

Основною дією для збільшення аудиторії, що користується нашим рішенням є проведення правильної маркетингової компанії окремо або у зв'язці із правильною реалізацією рішення у певній ситуації. Відтак можлива потреба у доробленні програмного забезпечення для використання актуальних технологій.

З можливих ризиків найвірогіднішими є вихід конкуренту у бюджетний сегмент ринку та зміна потреб користувачів. Обидва можуть бути ліквідовані описаними вище діями (табл. 4.6). Останні ж 3 ризики є менш вірогідними.

Також необхідно врахувати позитивні фактори, що можуть виникнути. При виникненні таких ситуацій завдяки правильним діям можна значно збільшити кількість клієнтів рішення. Можливі ситуації, що грають на руку проекту, наведені у таблиці 4.7.

Виходячи з проведеного аналізу можна сказати, що для охоплення всіх потенційних споживачів, необхідно забезпечити них хорошим зворотнім зв'язком, та створити зручне рішення для задоволення їх потреб.

Ринкові загрози – події, настання яких може несприятливо вплинути на підприємство.

Основною дією для збільшення аудиторії, що користується нашим рішенням є проведення правильної маркетингової компанії окремо або у зв'язці із правильною реалізацією рішення у певній ситуації. Відтак можлива потреба у доробленні програмного забезпечення для використання актуальних технологій.

Таблиця 4.7 - Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Зростання купувальних можливостей людей	Зростання середнього фінансового стану населення	Запропонувати їм свої послуги
2	Зниження довіри до конкурента	У конкурента вкрали інформацію із сховища	Проведення маркетингової компанії, що вказує на надійність рішення
3	Поява нових бібліотек	З'явилися нові бібліотеки, що покращують процес розробки та оптимізують рішення	Використання нової бібліотеки для розробки програмного рішення. Проведення маркетингової компанії, що вказує на підтримку інновацій рішенням
4	Зростаючий попит	Збільшення попиту на програми для прогнозування даних	Надавати високоякісні рішення, займати нішу ринку
5	Оптимізація швидкості роботи	Оптимізація швидкості роботи	Оптимізація швидкості роботи алгоритму за рахунок використання більш сучасних алгоритмів, індивідуального підбору алоритмів.

Для збільшення шансів виграти конкуренцію, необхідно провести ступеневий аналіз ринку. Це допоможе обрати стратегію поведінки на ринку та дозволить врахувати ключові особливості ринку. Результати аналізу зображено у таблиці 4.8.

Таблиця 4.8 - Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції: чиста.	Існує величезна кількість конкурентів на ринку.	Якісно провести рекламу.
2. За рівнем конкурентної боротьби: локальний.	На компанію впливатиме конкуренція в Україні.	Розробляти якісний продукт.
3. За галузевою ознакою: міжгалузева.	Конкуренти мають апаратно-програмне рішення, що може використовуватись лише всередині галузі.	Закласти у рішення можливість доробки для використання у інших галузях
4. Конкуренція за видами товарів: товарно-видова.	Послуги одного виду, але різної якості	Створити рішення враховуючи недоліки конкурентів та напрямок розвитку галузі

Продовження таблиці 4.8

5. За характером конкурентних переваг: цінова.	Цінова категорія сильно варіюється.	Враховувати ціни
6. За інтенсивністю: не марочна.	Значення мають технології і ціни, а не бренд.	Використовувати сучасні технології.

Проаналізувавши конкурентів та види конкурентної боротьби на ринку можна зробити висновок, що найважливішим фактором є сучасність рішення, його швидка доробка та зосередження уваги аудиторії на перевагах реалізації над виробами конкурентів. Також треба надати уваги створенню інтерфейсу різними мовами.

Необхідно виділити сильні позиції стартап-проекту у кожному з факторів: існуючі конкуренти, потенційні конкуренти, товари-замінники, постачальники, споживачі. Це допоможе оцінити привабливість реалізації. Результати аналізу (аналіз конкуренції в галузі за М. Портером) наведені у таблиці 4.9.

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти	Потенційні конкуренти	Постачальники	Клієнти	Замінники
	stackrox	Великі компанії, що створюють відділи для налаштування захисту в докер-контейнерах	Відсутні.	Важливою є точність та велика швидкодія.	Відсутні.
Висновки	Велика інтенсивність конкуренції.	Потенційно, при розширенні.	Постачальники не мають впливу.	Клієнти мають сильний вплив на роботу на ринку.	

Виходячи з аналізу можна сказати, що у проекту є можливості для входу на ринок. На ринку існують три конкуренти, найбільш схожою є реалізація конкуренту MS Excel. Існуючі постачальники диктують правила, проте наявність додаткового функціоналу дозволяє витримати конкуренцію. Основні вимоги користувачів покриваються даною реалізацією, тому можна сказати, що у проекту є шанси витримати конкуренцію.

На основі аналізу конкуренції на ринку, вимог, що ставляться

користувачами перед продуктом, та основними характеристиками ідеї проекту можна визначити та обґрунтувати основні фактори конкурентоспроможності рішення (представленні у таблиці 4.10).

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

№ п/ п	Фактор конкуренто- спроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Низька ціна рішення	Дозволяє охопити аудиторію, яка не може дозволити собі рішення конкурента
2	Використання відкритих бібліотек	Дозволяє знизити ціну розробки та підтримувати рішення актуальним, адже бібліотеки постійно доповнюються та оновлюються завдяки спеціалістам, що з ними працюють.
3	Зручність роботи з програмою	Дозволяє користувачу легко працювати з програмою
4	Кросплатформеність	Дозволяє запускати програму на різних платформах.

Можна сказати, що у стартап-проекті є достатньо факторів конкурентоспроможності, що надають йому переваги у боротьбі за споживачів. Також важливо відмітити, що низька ціна та відкритість у роботі із різними датчиками є ключовими побажаннями майбутніх користувачів, адже це надає більше свободи у виборі.

Надалі необхідно оцінити наскільки фактори конкурентоспроможності (табл. 10) та провести аналіз сильних та слабких сторін проекту. Основними сильними сторонами можна назвати низьку ціну та підтримку датчиків різних

виробників, адже реалізація конкурентів зроблена для роботи в інфраструктурі одного постачальника датчиків (табл. 4.11).

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін

№ п/п	Фактор конкуренто- спроможності	Бали 1- 20	Рейтинг товарів-конкурентів						
			-3	-2	-1	0	+1	+2	+3
1	Низька ціна рішення	15					✓		
2	Використання відкритих бібліотек	6		✓					
3	Зручність роботи з програмою	10				✓			
4	Кросплатформеність	17						✓	

Із результатів можна зробити висновок, що рішення має як і значні переваги, такі як кросплатформеність, так і свої недоліки, що проявляються у низькій кількості функціоналу порівняно із конкурентом. Проте фінальний продукт має бути конкурентоспроможним.

На основі проведеного раніше аналізу можна зробити SWOT аналіз факторів загроз, що можуть бути створенні як конкурентом, так і самим ринком, і аналіз факторів можливостей, що можуть виникнути у результаті помилок конкурентів, або в результаті змін складу цільової аудиторії (табл. 4.12).

Таблиця 4.12 – SWOT аналіз стартап-проекту

Сильні сторони: низька ціна, зручність у використанні.	Слабкі сторони: дуже насичений ринок, мала кількість функціоналу, відсутня кросплатформеність.
Можливості: насичення ринку новим підходом до прогнозування; різноманітна клієнтура.	Загрози: конкуренція.

На основі SWOT-аналізу розробляються альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок. Визначені альтернативи аналізуються з точки зору строків та імовірності отримання ресурсів.

Альтернативи ринкового впровадження стартап-проекту “Розроблення методів та засобів захисту докер-контейнерів інформаційно-освітнього середовища з використанням технологій шифрування даних ” наведено у Таблиці 4.13.

Таблиця 4.13 - Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Використання спеціалізованого рішення для захисту даних.	70%	4 місяців
2	Створення власного рішення з онлайн-інтеграцією.	60%	6 місяців

Отже можна зробити висновки: з означених альтернатив обирається та, для якої:

- а) отримання ресурсів є більш простим та імовірним;
- б) строки реалізації – більш стислими.

Оскільки у альтернативи 1 в більша вірогідність отримати кошти, і менші строки реалізації то буде логічним вибрати альтернативу 1.

4.4 Розробка ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (таблиця 4.14).

Виходячи із необхідностей різних цільових аудиторій, можна сказати, що характеристики фінальної реалізації найкраще підходять для людей, що працюють в областях машинного навчання та математичних розрахунків, оскільки для них важливо наявність унікальних особливостей, що допоможуть у розробці або автоматизації рутинних обчислень.

Таблиця 4.14 - Вибір цільових груп потенційних споживачів

№ п/ п	Опис профілю цільової групи потенційни х клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивніст ь конкуренції в сегменті	Простота входу у сегмент
1	Великі компанії.	Середня: велика конкуренція і можливість власних веб-відділів.	Середній.	Велика.	Важко.
2	Маленькі компанії.	Середня.	Низький.	Середня.	Середня.
3	Приватні особи.	Велика.	Високий.	Середня.	Легко .

Обрано цільову групу: Приватні особи.

За результатами аналізу потенційних груп споживачів (сегментів) обирають цільові групи, для яких пропонуватимуть товар, та визначають стратегію охоплення ринку:

- якщо компанія зосереджується на одному сегменті – вона обирає стратегію концентрованого маркетингу;
- якщо працює із кількома сегментами, розробляючи для них окремо програми ринкового впливу – вона використовує стратегію диференційованого маркетингу;
- якщо компанія працює із всім ринком, пропонуючи стандартизовану програму (включно із характеристиками товару/послуги) – вона використовує масовий маркетинг.

Обрано стратегію концентрованого маркетингу.

Визначення базової стратегії розвитку стартап-проекту “Мікросервісний підхід до розробки клієнтської частини веб-застосунків” наведено у Таблиці 4.15.

Таблиця 4.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Постійне оновлення і покращення продукту	Ринкове позиціонування на індивідуальних користувачів	Швидкодія, якість продукту	Концентрований маркетинг

Було обрано наступний варіант розвитку проекту: постійне оновлення і покращення продукту.

Стратегія лідера. Залежно від міри сформованості товарного (галузевого) ринку, характеру конкурентної боротьби компанії-лідери обирають одну з трьох стратегій: розширення первинного попиту, оборонну або наступальну стратегію або ж застосувати демаркетинг або диверсифікацію.

Стратегія розширення первинного попиту доцільна у разі, якщо фірмі-лідерові недоцільно розмінюватися на боротьбу з невеликими конкурентами, вона може отримати велику економічну віддачу від розширення первинного рівня попиту. В цьому випадку компанія займається реалізацією заходів по формуванню попиту (навчанню споживачів користуванню товаром, формування регулярного попиту, збільшення разового споживання), також пропаганду нових напрямів застосувань існуючих товарів, виявлень нових груп споживачів. У міру зростання ринку, його становлення позиції компанії-новатора починають атакувати конкуренти-імітатори. В цьому випадку, компанія може вибрати оборонну стратегію, метою якої є захист власної ринкової долі. Наступальна стратегія припускає збільшення своєї частки ринку. При цьому переслідувана мета полягає в подальшому підвищенні прибутковості роботи компанії на ринку за рахунок максимального використання ефекту масштабу. Якщо фірма потрапляє під дію антимонопольного законодавства, вона може удатися до стратегії демаркетинга, що припускає скорочення своєї частки ринку, зниження рівня попиту на деяких сегментах за рахунок підвищення ціни. При цьому ставиться завдання недопущення на ці сегменти конкурентів, а компенсація втрат прибутку через зменшення обсягів виробництва компенсується встановленням надвисоких цін.

Стратегія виклик у лідера. Стратегію виклику лідерові найчастіше вибирають компанії, які є другими, третіми на ринку, але бажають стати лідером ринку. Теоретично, ці компанії можуть прийняти два стратегічні рішення: атакувати лідера у боротьбі за частку ринку або ж йти за лідером. Рішення

атакувати лідера є досить ризикованим. Для його реалізації потрібні значні фінансові витрати, know – how, краще співвідношення «ціна- якість», переваги в системі розподілу і просування і т. д. У разі не реалізації цієї стратегії, компанія може бути відкинута на аутсайдерські позиції на досить довгий час. Залежно від цього компанія може вибрати одну з альтернативних стратегій: фронтальної або флангової атаки. Стратегія наслідування лідер у . Компанії, що приймають слідування за лідером – це підприємства з невеликою часткою ринку, які вибирають адаптивну лінію поведінки на ринку, усвідомлюють своє місце на ній і йдуть у фарватері фірм-лідерів. Головна перевага такої стратегії – економія фінансових ресурсів, пов'язаних з необхідністю розширення товарного(галузевого) ринку, постійними інноваціями, витратами на утримання домінуючого положення.

Стратегія заняття конкурентної ніші. При прийнятті стратегії зайняття конкурентної ніші (інші назви – стратегія фахівця або нішера) компанія в якості цільового ринку вибирає один або декілька ринкових сегментів. Головна особливість – малий розмір сегментів/сегменту. Ця конкурентна стратегія являється похідною від такої базової стратегії компанії, як концентрація. Головне завдання для компаній, що вибирають стратегію нішера або фахівця, – це постійна турбота про підтримку і розвиток своєї конкурентної переваги, формування лояльності і прихильності споживачів, підтримка вхідних бар'єрів.

У таблиці 4.16 наведено базову стратегію конкурентної поведінки.

Таблиця 4.16 - Визначення базової стратегії конкурентної поведінки

№ п / п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія Конкурентної поведінки
1	Ні.	Компанія буде шукати нових споживачів та забирати існуючих у конкурентів.	Буде копіювати та удосконалювати	Зайняття конкурентної ніші

Продукт не є першопроходьцем, адже конкуренти уже існують на ринку, і рішення буде частково скопійованим у них, проте буде мати свої особливості. Проте оскільки ринок розвиваючийся, то кількість нових клієнтів, що лише шукають рішення для себе досить велика, тому компанія буде шукати як нових клієнтів, так і відбивати клієнтів конкурента. Через це найкращим варіантом конкурентної поведінки буде зайняття конкурентної ніші, адже ця ніше задовольняє таким умовам: є досить прибутковою і реалізація намагається зайняти малу нішу не підприємницького сегменту.

Визначимо стратегію позиціонування у таблиці 4.17, що полягає у формуванні ринкової позиції (комплексу асоціації), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 4.17 - Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап- проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1.	Низька ціна, простий інтерфейс	Диференціація та спеціалізація	Спеціалізація на потребах клієнтів, швидке впровадження нового функціоналу.	Дешевизна, простота, ефективність

Вимоги цільової аудиторії співпадають із основними конкурентними якостями проекту. Стратегія розвитку полягає у спеціалізації споживачів і постійному впровадженні нових функцій, які вони потребують.

4.5 Розробка маркетингової програми стартап-проекту

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього у табл. 4.18 потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару.

Таблиця 4.18 - Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1.	Низька ціна	Товар має найнижчу вартість на ринку, проте забезпечується хороший зворотній зв'язок із виробником	Нижча ціна
2.	Простота користувацького інтерфейсу	Простота мови виразів.	Користувачі мають інтерфейс, що більш зручний.

Реалізація задовольняє основні потреби аудиторій, тим саме отримуючи конкурентну перевагу. Зв'язок реалізований за допомогою RESTfull API допоможе стандартизувати методи доступу до інформації та полегшить розробку, завдяки використанню широкорозповсюдженної технології.

Далі у таблиці 4.19 проілюстрована трирівнева маркетингова модель товару: уточняється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання.

Таблиця 4.19 - Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
1.Товар за задумом	Програма для прогнозування часових рядів		
2. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Індивідуальний підхід.	1.Нм	1.Технологічна
	2. Низька ціна.	2.Нм	2.Економічна
	3. Простота у використанні.	3.Нм	3.Технологічна
	Якість: згідно зі стандартом ISO 4444 буде проведено тестування.		
	Маркування відсутнє.		
Моя компанія: “Flandre Team”.			
3. Товар із підкріпленням	Постійна підтримка для користувачів.		
За рахунок чого потенційний товар буде захищено від копіювання: ліцензія.			

Було розглянуто три рівні моделі товару, з чого можна зробити висновок, що властивості є як економічні та матеріальні, так і технологічні і нематеріальні. Також було надано сутність та складові товару у задумці та товару з підкріпленням. Після формування маркетингової моделі товару слід особливо відмітити – чим саме проект буде захищено від копіювання. У даному випадку найбільш вірогідним гарантом буде ліцензія.

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів

цільової групи споживачів (табл. 4.20). Аналіз проводиться експертним методом.

Таблиця 4.20 - Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1.	0 - 14000 грн	0 - 15000 грн	>100000 грн/рік	4500-11000 грн

Далі необхідно визначити основні системи збуту, в межах яких приймається рішення (табл. 4.21).

Таблиця 4.21 - Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Купують одноразово готове рішення	Продаж	0(напрямую)	Власна

Система буде приносити прибуток за рахунок постійного притоку клієнтів.

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 4.22).

Таблиця 4.22 - Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1.	Встановлення програми у персональній комп'ютер і його використання	Інтернет	Низька ціна, простота використання, універсальність	Показати переваги рішення над конкурентами, виділити ключові особливості	Створення сайту продукту, розповсюдження інформації про продукт на спеціалізованих ресурсах.

Було визначено, що придбання продукту буде проводитись через мережу Інтернет або при безпосередньому спілкуванні із представниками компанії. Розповсюдження інформації про продукт буде проводитись виключно через Інтернет, адже аудиторія даного продукту активно користується всесвітньою мережею.

4.6 Висновки

Дослідження у цьому розділі показало, що проект можна комерціалізувати, не зважаючи на те, що ринок має своїх монополістів та свої правила. Хоча реалізація має ряд недоліків перед конкурентами, наявні переваги. Для успішного виходу на ринок, потрібно, щоб:

- проект мав низьку собівартість;
- компанія-розробник налагодила зворотній зв'язок із користувачами та впроваджувала новий функціонал;
- при рекламі продукту акцентувалась увага на сильних сторонах продукту.

Було проведено аналіз потенційних ризиків і можливостей, а також розраховані основні фінансово-економічні показники проекту. Отримані результати кажуть про те, що реалізація проекту є доцільною.

Було визначено сильні сторони проекту: зручність у використанні, ціна, точність. Серед слабких варто виділити невелику кількість функціоналу.

Варто відмітити можливість реклами продукту на спеціалізованих ресурсах із зазначенням сильних сторін проекту.

ВИСНОВКИ

Точне прогнозування фондових ринків є складним завданням, оскільки існують мільйони подій та передумов для того, щоб акціонерні товариства рухались у певному напрямку. У цій роботі було запропоновано просту у використанні модель прогнозування фондового, щоб допомогти дедалі більшій кількості нефінансових професійних звичайних інвесторів приймати рішення. Було використано навчання підкріпленням при оптимізації гіперпараметрів, адже фондові ринки весь час змінюються. Навіть якщо вдається навчити GAN та LSTM створювати надзвичайно точні результати, результати можуть бути дійсними лише протягом певного періоду. Це означає, що потрібно постійно оптимізувати весь процес. Заснована на мережі глибокого навчання, ця модель досягає здатності прогнозування, що перевершує інші базові методи, за допомогою змагальних тренувань, мінімізації втрат прогнозування напрямку та втрати прогнозованих помилок. Більше того, аналізуються ефекти циклів оновлення моделі на можливість прогнозування, а експериментальні результати показують, що менший цикл оновлення моделі може отримати кращі результати прогнозування. У майбутньому ми спробуємо створити середовище RL для тестування торгових алгоритмів, які вирішують, коли і як торгувати. Вихід з GAN буде одним із параметрів у середовищі.

ПЕРЕЛІК ПОСИЛАНЬ

1. R. Al-Hmouz, W. Pedrycz, and A. Balamash. Description and prediction of time series: a general framework of Granular Computing. *Expert Systems with Applications*. 2015. Vol. 42, No. 10. P. 4830–4839.
2. S. Barak and M. Modarres. Developing an approach to evaluate stocks by forecasting effective features with data mining methods. *Expert Systems with Applications*. 2015. Vol. 42, No. 3. H. 1325–1339.
3. A. Booth, E. Gerding, and F. McGroarty. Automated trading with performance weighted random forests and seasonality. *Expert Systems with Applications*. 2014. Vol. 41, No. 8. P. 3651–3661.
4. A. Bagheri, H. Mohammadi Peyhani, and M. Akbari. Financial forecasting using ANFIS networks with Quantum-behaved Particle Swarm Optimization. *Expert Systems with Applications*. 2014. Vol. 41, No. 14. P. 6235–6250.
5. Y. Son, D.-J. Noh, and J. Lee. Forecasting trends of high-frequency KOSPI200 index data using learning classifiers. *Expert Systems with Applications*. 2012. Vol. 39, No. 14. P. 11607–11615.
6. I. Aldridge and S. Krawciw. Real-Time Risk: What Investors Should Know About FinTech. in *High-Frequency Trading, and Flash Crashes*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2017. P.67-79.
7. F. A. De Oliveira, C. N. Nobre, and L. E. Zárate. Applying Artificial Neural Networks to prediction of stock price and improvement of the directional prediction index. Case study of PETR4, Petrobras, Brazil, *Expert Systems with Applications*. 2013 Vol. 40, No. 18. P. 7596–7606.
8. J. H. Niño-Peña and G. J. Hernández-Pérez. Price direction prediction on high frequency data using deep belief networks. *Communications in Computer and Information Science*. 2016. Vol. 657, P. 74–83.
9. A. Marszałek and T. Burczyn'ski. Modeling and forecasting financial time series with ordered fuzzy candlesticks. *Information Sciences*. 2014. Vol. 273, P. 144–155.

- 10.X. Li, X. Huang, X. Deng, and S. Zhu. Enhancing quantitative intra-day stock return prediction by integrating both market news and stock prices information. *Neurocomputing* 2014. Vol. 142, P. 228–238. 2014.
- 11.X. Wang, S. Bao, and J. Chen. High-frequency stock linkage and multi-dimensional stationary processes. *Physica A: Statistical Mechanics and its Applications*. 2017. Vol. 468, P. 70–83.
- 12.E. Chong, C. Han, and F. C. Park. Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications*. 2017. Vol. 83, P. 187–205.
- 13.I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al.. Generative adversarial nets. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems 2014, NIPS 2014*. P. 2672–2680.
- 14.S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics*. 2017. Vol. 36, No. 4. Article no. 107.
- 15.P. Luc, C. Couprie, S. Chintala, and J. Verbeek. *Semantic segmentation using adversarial networks, arXiv preprint*. ArXiv, 1611.08408, 2016, arXiv:1611.08408.
- 16.M. Mathieu, C. Couprie, and Y. LeCun. *Deep multi-scale video prediction beyond mean square error, arXiv preprint*. ArXiv, 1511.05440, 2015, arXiv:1511.05440.
- 17.J. D. Hamilton. *Time Series Analysis*. New Jersey: Princeton University Press, 1994. 799 p.
- 18.R. H. Shumway and D. S. Stoffer. *Time series analysis and its applications*, Springer Texts in Statistics 3rd edition. New York, NY, USA: Springer, 2011. 562 p.
- 19.P. J. Brockwell and R. A Davis. *Time Series: Theory and Methods*. New York, NY, USA: Springer Science & Business Media .2013. 519 p.
- 20.S. Pellegrini, E. Ruiz, and A. Espasa. Prediction intervals in conditionally heteroscedastic time series with stochastic components. *International Journal of Forecasting*. 2011. Vol. 27, No. 2. P. 308–319.

21. Y. Kara, M. Acar Boyacioglu, and Ö. K. Baykan. Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. *Expert Systems with Applications*. 2011. Vol. 38, No. 5. P. 5311–5319.
22. M. Ghiassi, J. Skinner, and D. Zimbra. Twitter brand sentiment analysis: a hybrid system using n-gram analysis and dynamic artificial neural network. *Expert Systems with Applications*. 2013. Vol. 40, No. 16. P. 6266–6282.
23. M. R. Hassan. A combination of hidden Markov model and fuzzy model for stock market forecasting. *Neurocomputing*. 2009. Vol. 72, No. 16-18. P. 3439–3446.
24. W. Huang, Y. Nakamori, and S.-Y. Wang. Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*. 2005. Vol. 32, No. 10. P. 2513–2522.
25. A. F., S. Elsir, and H. Faris. A Comparison between Regression, Artificial Neural Networks and Support Vector Machines for Predicting Stock Market Index. *International Journal of Advanced Research in Artificial Intelligence*. 2015. Vol. 4, No. 7. P. 562 – 571.
26. R. Majhi, G. Panda, G. Sahoo, A. Panda, and A. Choubey. Prediction of S&P 500 and DJIA stock indices using particle swarm optimization technique. In *Proceedings of the Proceeding of the IEEE Congress on Evolutionary Computation (CEC '08)*. P. 1276–1282, Hong Kong, China, June 2008.
27. S. S. Appadoo. *Pricing Financial Derivatives with Fuzzy Algebraic Models: A Theoretical and Computational Approach [Ph.D. thesis]*. University of Manitoba, Winnipeg, 2006. P. 5321-5328.
28. A. Thavaneswaran, K. Thiagarajah, and S. S. Appadoo. Fuzzy coefficient volatility ({FCV}) models with applications. *Mathematical and Computer Modelling*. 2007. Vol. 45, No. 7-8. P. 777–786.
29. C. Carlsson and R. Fullér. On possibilistic mean value and variance of fuzzy numbers. *Fuzzy Sets and Systems*. 2001. Vol. 122, No. 2. P. 315–326.
30. A. Thavaneswaran, S. S. Appadoo, and A. Paseka. Weighted possibilistic moments of fuzzy numbers with applications to {GARCH} modeling and option pricing. *Mathematical and Computer Modelling*. 2009. Vol. 49, No. 1-2. P. 352–368.

31. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*. 2014. Vol. 15, No. 1. P. 1929–1958.
32. A. M. Rather, A. Agarwal, and V. N. Sastry. Recurrent neural network and a hybrid model for prediction of stock returns. *Expert Systems with Applications*. 2015. Vol. 42, No. 6. P. 3234–3241.
33. R. D. A. Araújo, A. L. I. Oliveira, and S. Meira. A hybrid model for high-frequency stock market forecasting. *Expert Systems with Applications*. 2015. Vol. 42, No. 8. P. 4081–4096.
34. H. Chen, K. Xiao, J. Sun, and S. Wu. A double-layer neural network framework for high-frequency forecasting. *ACM Transactions on Management Information Systems (TMIS)*. 2017. Vol. 7, No. 4. Article no. 11.
35. A. Yoshihara, K. Fujikawa, K. Seki, and K. Uehara. Predicting Stock Market Trends by Recurrent Deep Neural Networks. In *PRICAI 2014: Trends in Artificial Intelligence*. Vol. 8862 of *Lecture Notes in Computer Science*. P. 759–769, Springer International Publishing, Cham, 2014.
36. X. Ding, Y. Zhang, T. Liu, and J. Duan. Deep learning for event-driven stock prediction. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence, IJCAI 2015*. P. 2327–2333, arg, July 2015.
37. Z. Zhou, J. Zhao, and K. Xu. Can online emotions predict the stock market in China. In *Proceedings of the International Conference on Web Information Systems Engineering*. 2016. P. 328–342.

ДОДАТОК А ВХІДНІ ДАНІ

6316,17686	6344,56851	6653,29053	6901,50393	6854,41735
6293,64172	6393,88816	6676,6314	6760,28988	6750,54208
6258,35865	6383,65217	6677,93554	6495,91887	6811,03993
6339,14052	6394,6665	6662,65828	6665,97888	6881,28304
6314,51034	6389,90685	6708,49226	6582,02252	6913,02195
6308,6086	6466,32037	6758,54098	6306,09995	6929,38526
6378,62808	6513,26837	6737,13714	6412,68031	6966,42761
6386,11943	6480,66761	6810,28339	6523,8513	7101,18023
6409,2867	6472,48247	6811,37811	6553,86428	7131,1209
6405,96553	6472,68766	6834,32938	6675,03153	7046,51113
6422,55733	6465,16566	6906,27807	6794,92198	7040,98089
6311,37498	6433,16201	6963,46297	6770,66346	7030,96966
6365,56374	6435,15211	6919,35245	6779,69491	7019,9489
6337,874	6441,42052	6916,30432	6759,26006	6864,88225
6263,70123	6396,42217	7022,97118	6761,85464	6885,91979
6265,11096	6511,3424	6988,31543	6896,6012	6853,45209
6293,05353	6575,80269	6930,72732	6989,09712	6682,25518
6316,2778	6584,5778	6949,98684	6900,35042	6508,08922

6753,83223	6509,04861	6893,6201	7280,70466	7352,35749
6529,84122	6513,94051	6953,63071	7217,48814	7350,22689
6460,81022	6649,65367	6949,69859	7197,51188	7371,78014
6581,12622	6656,35329	6960,91879	7038,17063	7406,24779
6390,83694	6605,56959	6926,54391	7068,19541	7508,58635
6458,83109	6681,96065	6976,3668	6969,67364	7400,74811
6560,05967	6644,47827	6967,72943	7031,60387	7296,78093
6594,83815	6643,47932	7083,92519	7040,80169	7193,10289
6433,2132	6769,11597	7143,57017	7097,81947	7231,98322
6472,34441	6821,8738	7166,75239	7014,5543	7272,88738
6615,87245	6815,48258	7210,08171	7101,04682	7372,14808
6583,44381	6893,21419	7152,83023	7207,33383	7395,48852
6656,25508	6963,54765	7152,62091	7276,00141	7438,9947
6628,34237	6952,56314	7168,47891	7282,59897	7462,65309
6675,17875	6964,3727	7209,18282	7243,97977	7469,54496
6816,36863	6888,53577	7205,25573	7366,25024	7466,96198
6833,2127	6929,96538	7279,59195	7375,81651	7408,30398
6774,88747	6901,63254	7255,75585	7357,89858	7401,16941
6667,75096	6866,25394	7251,40756	7403,89139	7447,16789
6648,80306	6905,53226	7228,04343	7390,12595	7354,66126

7374,29478	7545,49631	7157,20854	7039,14917	6704,23669
7377,54416	7434,72945	7068,67059	6829,09764	6763,9567
7371,41785	7494,39938	7276,42632	6830,90574	6767,97351
7397,22571	7490,32306	7278,63114	6769,87257	6594,96322
7424,59561	7569,03455	7116,08832	6890,44702	6448,38643
7413,83805	7531,07186	7107,23298	6867,01533	6491,51633
7485,40405	7548,75069	7141,21265	6642,9165	6342,96811
7559,13289	7563,19385	7118,67447	6526,9629	6243,19479
7570,24551	7563,08645	6789,15431	6575,663	6046,55657
7660,17967	7629,57243	7016,38934	6527,34732	5899,35464
7642,66991	7627,64992	6852,40434	6678,33898	6262,76658
7654,55356	7645,45309	6713,90178	6701,03621	6288,30198
7622,32237	7628,27827	6810,12107	6913,32614	6285,26637
7523,25959	7637,4258	6967,09572	6892,39969	6329,96459
7453,16718	7490,0036	7069,17116	6949,01388	6360,86916
7430,26067	7399,00713	6965,29401	7062,13316	6147,12763
7447,67609	7352,82313	6937,08939	6795,21336	6422,67373
7507,87224	7371,61536	6988,84763	6838,84985	6488,25357
7488,06459	7044,49552	7203,133	6613,27499	6551,84732
7561,68654	6964,02648	7158,59078	6682,74232	6600,69437

6620,94166	6913,12991	7164,02044	7599,74238	7640,14914
6601,39529	6909,18472	7201,28133	7568,48632	7617,54964
6541,03773	7014,66903	7256,98152	7611,48785	7582,74635
6669,63803	7015,87582	7243,01359	7594,88997	7586,53111
6668,56199	7022,41911	7306,98721	7628,14977	7324,13217
6718,45304	7055,17934	7326,2815	7629,11979	7401,88367
6784,60784	7066,61207	7349,27825	7654,73136	7503,24563
6646,81301	7062,33851	7380,75372	7680,71752	7580,14232
6658,76341	7035,16399	7493,26939	7689,71511	7503,67966
6702,54694	7090,62505	7326,05666	7713,49485	7376,70336
6787,37138	7115,42606	7316,95771	7810,70617	7451,01961
6697,08682	7123,21568	7351,15018	7784,41258	7420,65614
6632,79238	7117,0046	7308,1931	7816,91905	7307,93346
6807,90886	7097,5255	7320,4723	7826,67914	7300,96079
6906,83934	7151,57205	7378,77147	7839,03833	7278,37589
6875,5179	7150,83335	7478,41559	7781,4557	7216,8587
6959,95692	7156,78508	7499,63884	7751,84691	7245,40294
7023,52079	7112,4677	7544,97153	7724,05617	7127,95613
6997,6223	7026,88267	7540,56762	7845,72919	6978,01746
6904,98065	7015,69004	7578,83998	7794,09288	7166,74739

7220,90316	7857,68984	7801,14635	7702,3117	7681,57853
7275,92674	7841,30105	7692,80054	7690,99781	7749,44866
7417,29126	7785,78668	7415,69073	7609,51316	7684,14216
7501,9289	7826,85832	7521,32187	7719,24644	7550,78612
7513,84912	7903,4007	7551,89983	7862,53913	7638,39457
7472,2944	7896,77579	7724,82982	7852,53856	7754,10247
7510,68343	7943,24114	7646,26782	7832,40322	7725,12947
7479,11145	7966,92622	7561,68245	7814,74229	7604,26972
7526,51939	7927,07708	7728,15145	7887,58059	7690,5292
7635,40014	7888,76195	7490,13037	7917,3398	7740,36226
7667,73845	7904,13215	7484,88791	7892,95465	7843,87499
7738,05471	7834,89676	7604,10877	7852,41238	7842,33373
7728,78158	7905,11927	7719,32331	7888,7863	7942,8509
7723,02146	7954,56425	7664,47094	7888,55621	7920,20882
7591,53633	8010,60499	7733,21575	7901,79211	7942,14354
7627,05087	7929,86678	7707,42711	7823,55329	7868,49078
7657,04832	8016,95277	7464,9945	7818,61003	7940,33087
7671,07487	7989,0819	7575,01863	7710,04445	7874,6156
7768,13825	7952,47269	7566,0273	7803,54135	7889,47261
7799,82355	7848,77978	7587,8962	7771,99214	7966,71865

8029,21718	8161,16696	8255,88592	8315,5231	8272,05251
8110,66907	8210,66448	8241,91184	8328,47885	8371,92798
8047,50863	8210,17532	8263,78873	8338,73731	8385,74521
8083,11257	8196,02791	8259,81315	8283,75293	8444,70885
8083,83238	8219,64637	8257,83079	8265,61816	8403,68487

ДОДАТОК Б ЛІСТІНГ ПРОГРАМИ

```

class TriangularSchedule():
    def __init__(self, min_lr, max_lr, cycle_length, inc_fraction=0.5):
        self.min_lr = min_lr
        self.max_lr = max_lr
        self.cycle_length = cycle_length
        self.inc_fraction = inc_fraction

    def __call__(self, iteration):
        if iteration <= self.cycle_length*self.inc_fraction:
            unit_cycle = iteration * 1 / (self.cycle_length * self.inc_fraction)
        elif iteration <= self.cycle_length:
            unit_cycle = (self.cycle_length - iteration) * 1 / (self.cycle_length * (1 -
self.inc_fraction))
        else:
            unit_cycle = 0
        adjusted_cycle = (unit_cycle * (self.max_lr - self.min_lr)) + self.min_lr
        return adjusted_cycle

class CyclicalSchedule():
    def __init__(self, schedule_class, cycle_length, cycle_length_decay=1,
cycle_magnitude_decay=1, **kwargs):
        self.schedule_class = schedule_class
        self.length = cycle_length
        self.length_decay = cycle_length_decay
        self.magnitude_decay = cycle_magnitude_decay
        self.kwargs = kwargs

    def __call__(self, iteration):
        cycle_idx = 0
        cycle_length = self.length
        idx = self.length
        while idx <= iteration:
            cycle_length = math.ceil(cycle_length * self.length_decay)
            cycle_idx += 1
            idx += cycle_length
        cycle_offset = iteration - idx + cycle_length

        schedule = self.schedule_class(cycle_length=cycle_length, **self.kwargs)
        return schedule(cycle_offset) * self.magnitude_decay**cycle_idx

schedule = CyclicalSchedule(TriangularSchedule, min_lr=0.5, max_lr=2, cycle_length=500)
iterations=1500

plt.plot([i+1 for i in range(iterations)], [schedule(i) for i in range(iterations)])
plt.title('Learning rate for each epoch')
plt.xlabel("Epoch")
plt.ylabel("Learning Rate")
plt.show()

```

```

gan_num_features = dataset_total_df.shape[1]
sequence_length = 17

class RNNModel(gluon.Block):
    def __init__(self, num_embed, num_hidden, num_layers, bidirectional=False, \
                  sequence_length=sequence_length, **kwargs):
        super(RNNModel, self).__init__(**kwargs)
        self.num_hidden = num_hidden
        with self.name_scope():
            self.rnn = rnn.LSTM(num_hidden, num_layers, input_size=num_embed, \
                                bidirectional=bidirectional, layout='TNC')

            self.decoder = nn.Dense(1, in_units=num_hidden)

    def forward(self, inputs, hidden):
        output, hidden = self.rnn(inputs, hidden)
        decoded = self.decoder(output.reshape((-1, self.num_hidden)))
        return decoded, hidden

    def begin_state(self, *args, **kwargs):
        return self.rnn.begin_state(*args, **kwargs)

lstm_model = RNNModel(num_embed=gan_num_features, num_hidden=500, num_layers=1)
lstm_model.collect_params().initialize(mx.init.Xavier(), ctx=mx.cpu())
trainer = gluon.Trainer(lstm_model.collect_params(), 'adam', {'learning_rate': .01})
loss = gluon.loss.L1Loss()

n_epoch = 150
print_period = n_epoch // 10
start = time.time()

training_loss = []
validation_loss = []
for epoch in range(n_epoch):
    epoch_loss = 0
    epoch_val_loss = 0

    train_iter.reset()
    test_iter.reset()

    n_batch_train = 0
    for batch in train_iter:
        n_batch_train += 1
        data = batch.data[0].as_in_context(mx.cpu())

        with autograd.record():
            loss = net(data)
        loss.backward()
        trainer.step(data.shape[0])
        epoch_loss += nd.mean(loss).asscalar()

```

```

n_batch_val = 0
for batch in test_iter:
    n_batch_val += 1
    data = batch.data[0].as_in_context(mx.cpu())
    loss = net(data)
    epoch_val_loss += nd.mean(loss).asscalar()

epoch_loss /= n_batch_train
epoch_val_loss /= n_batch_val

training_loss.append(epoch_loss)
validation_loss.append(epoch_val_loss)

"""if epoch % max(print_period, 1) == 0:
    print('Epoch {}, Training loss {:.2f}, Validation loss {:.2f}'.\
          format(epoch, epoch_loss, epoch_val_loss))"""

end = time.time()
print('Training completed in {} seconds.'.format(int(end-start)))

batch_size = 64
n_batches = VAE_data.shape[0]/batch_size
VAE_data = VAE_data.values

train_iter = mx.io.NDArrayIter(data={'data': VAE_data[:num_training_days,:-1]}, \
                                label={'label': VAE_data[:num_training_days, -1]},
                                batch_size=batch_size)
test_iter = mx.io.NDArrayIter(data={'data': VAE_data[num_training_days,:-1]}, \
                               label={'label': VAE_data[num_training_days, -1]},
                               batch_size=batch_size)
model_ctx = mx.cpu()
class VAE(gluon.HybridBlock):
    def __init__(self, n_hidden=400, n_latent=2, n_layers=1, n_output=784, \
                 batch_size=100, act_type='relu', **kwargs):
        self.soft_zero = 1e-10
        self.n_latent = n_latent
        self.batch_size = batch_size
        self.output = None
        self.mu = None
        super(VAE, self).__init__(**kwargs)

    with self.name_scope():
        self.encoder = nn.HybridSequential(prefix='encoder')

        for i in range(n_layers):
            self.encoder.add(nn.Dense(n_hidden, activation=act_type))
        self.encoder.add(nn.Dense(n_latent*2, activation=None))

        self.decoder = nn.HybridSequential(prefix='decoder')
        for i in range(n_layers):
            self.decoder.add(nn.Dense(n_hidden, activation=act_type))

```

```

        self.decoder.add(nn.Dense(n_output, activation='sigmoid'))

    def hybrid_forward(self, F, x):
        h = self.encoder(x)
        #print(h)
        mu_lv = F.split(h, axis=1, num_outputs=2)
        mu = mu_lv[0]
        lv = mu_lv[1]
        self.mu = mu

        eps = F.random_normal(loc=0, scale=1, shape=(self.batch_size, self.n_latent),
                               ctx=model_ctx)
        z = mu + F.exp(0.5*lv)*eps
        y = self.decoder(z)
        self.output = y

        KL = 0.5*F.sum(1+lv-mu*mu-F.exp(lv),axis=1)
        logloss = F.sum(x*F.log(y+self.soft_zero)+ (1-x)*F.log(1-y+self.soft_zero),
                        axis=1)
        loss = -logloss-KL

        return loss

n_hidden=400 # neurons in each layer
n_latent=2
n_layers=3 # num of dense layers in encoder and decoder respectively
n_output=VAE_data.shape[1]-1

net = VAE(n_hidden=n_hidden, n_latent=n_latent, n_layers=n_layers, n_output=n_output,
          batch_size=batch_size, act_type='gelu')
net.collect_params().initialize(mx.init.Xavier(), ctx=mx.cpu())
net.hybridize()
trainer = gluon.Trainer(net.collect_params(), 'adam', {'learning_rate': .01})
print(net)

from pandas import read_csv
from pandas import datetime
from statsmodels.tsa.arima_model import ARIMA
from sklearn.metrics import mean_squared_error

X = series.values
size = int(len(X) * 0.66)
train, test = X[0:size], X[size:len(X)]
history = [x for x in train]
predictions = list()
for t in range(len(test)):
    model = ARIMA(history, order=(5,1,0))
    model_fit = model.fit(dispatch=0)
    output = model_fit.forecast()
    yhat = output[0]
    predictions.append(yhat)
    obs = test[t]

```



```

        history.append(obs)
    error = mean_squared_error(test, predictions)
    print('Test MSE: %.3f' % error)
    def get_feature_importance_data(data_income):
        data = data_income.copy()
        y = data['price']
        X = data.iloc[:, 1:]

        train_samples = int(X.shape[0] * 0.65)

        X_train = X.iloc[:train_samples]
        X_test = X.iloc[train_samples:]

        y_train = y.iloc[:train_samples]
        y_test = y.iloc[train_samples:]

        return (X_train, y_train), (X_test, y_test)
    # Get training and test data
    (X_train_FI, y_train_FI), (X_test_FI, y_test_FI) =
    get_feature_importance_data(dataset_TI_df)
    regressor =
    xgb.XGBRegressor(gamma=0.0, n_estimators=150, base_score=0.7, colsample_bytree=1, learning_r
    ate=0.05)
    xgbModel = regressor.fit(X_train_FI, y_train_FI, \
                             eval_set = [(X_train_FI, y_train_FI), (X_test_FI, y_test_FI)],
    \
                             verbose=False)
    eval_result = regressor.evals_result()
    training_rounds = range(len(eval_result['validation_0']['rmse']))
    plt.figure(figsize=(15, 5))
    plt.subplots_adjust(left=None, bottom=None, right=None, top=None, wspace=.5,
    hspace=None)

    ranges_ = (-10, 3, .25)

    plt.subplot(1, 2, 1)
    plt.plot([i for i in np.arange(*ranges_)], [relu(i) for i in np.arange(*ranges_)],
    label='ReLU', marker='.')
    plt.plot([i for i in np.arange(*ranges_)], [gelu(i) for i in np.arange(*ranges_)],
    label='GELU')
    plt.hlines(0, -10, 3, colors='gray', linestyle='--', label='0')
    plt.title('Figure 7: GELU as an activation function for autoencoders')
    plt.ylabel('f(x) for GELU and ReLU')
    plt.xlabel('x')
    plt.legend()

    plt.subplot(1, 2, 2)
    plt.plot([i for i in np.arange(*ranges_)], [lrelu(i) for i in np.arange(*ranges_)],
    label='Leaky ReLU')
    plt.hlines(0, -10, 3, colors='gray', linestyle='--', label='0')
    plt.ylabel('f(x) for Leaky ReLU')
    plt.xlabel('x')
    plt.title('Figure 8: LeakyReLU')
    plt.legend()

```

```
plt.show()
# Plot the predicted (from ARIMA) and real prices

plt.figure(figsize=(12, 6), dpi=100)
plt.plot(test, label='Real')
plt.plot(predictions, color='red', label='Predicted')
plt.xlabel('Days')
plt.ylabel('USD')
plt.title('Figure 5: ARIMA model on GS stock')
plt.legend()
plt.show()
```